



UNIVERSIDADE SALVADOR - UNIFACS  
PROGRAMA DE PÓS-GRADUAÇÃO EM REDES DE COMPUTADORES  
MESTRADO PROFISSIONALIZANTE EM REDES DE COMPUTADORES

**MANOEL CARVALHO MARQUES NETO**

**GRAPHMINER: UMA FERRAMENTA PARA  
VISUALIZAÇÃO DE DADOS EM BANCOS  
DE DADOS RELACIONAIS**

Salvador  
26 de Abril de 2004

Manoel Carvalho Marques Neto

## **GRAPHMINER: UMA FERRAMENTA PARA VISUALIZAÇÃO DE DADOS EM BANCOS DE DADOS RELACIONAIS**

*Dissertação apresentada à Universidade Salvador, como parte das exigências do Curso de Mestrado Profissional em Redes de Computadores, área de concentração em Redes de Computadores, para obtenção do título de Mestre.*

Orientador: *Prof. Dr. Manoel Gomes de Mendonça Neto*

Co-orientador: *Prof. Dr. Celso Alberto Santos Saibel*

Salvador  
26 de Abril de 2004

*Dedico este trabalho aos meus Pais, Gessé e Rosa, aos meus Irmãos Gessezinho, Lara, Lidia, Cris, Rosinha, Dai, Marlene e a minha Avó "Cotinha".*

# AGRADECIMENTOS

Ufa!!! Terminei ... Pensei que não chegaria ao fim... Quem inventou esse negócio de mestrado devia ter lembrado de reservar um tempo para namorar, viajar, assistir novela e outras coisas menos estressantes!!! .... Devia ter escutado minha irmã Lídia e ter feito medicina.

A lista de agradecimentos é meio extensa por isso vou tentar resumir. Vou começar pela família. Agradeço ao meu querido pai GESSÉ e a minha mãe ROSA pelo esforço e dedicação em todos esses anos de vida....É uma pena que o senhor não esteja aqui “Painho” para presenciar esse momento.....Sei que ficaria muito feliz....

Agradeço, imensamente, aos meus irmãos (LARA, GESSÉ, CRISTIANE, ROSINHA, MARLENE e DAI) pelo apoio depois que nosso pai partiu. Dedico um agradecimento especial para minha irmã, mãe e pai LÍDIA, cujo empenho pessoal foi imprescindível para que eu me tornasse um homem adulto. Apesar de todas as nossas discordâncias você sabe que eu te amo muito!!! Compartilho com todos vocês os méritos alcançados

Agradecimentos especiais aos meus dois professores orientadores pelos conhecimentos compartilhados e pela compreensão, aos professores VICENTE, ISAAC e EUCLÉRIO que desde a época da graduação sempre incentivaram esse meu sonho, aos colegas de turma, e do laboratório (não vou arriscar esquecer de alguém!!!!) especialmente a minha amiga CAROL PASSOS, que sempre dividiu comigo todas as alegrias e tristezas desses últimos dois anos. Um agradecimento especial aos meus queridos alunos LILE e GUIDO. Vocês todos foram muito importantes para que eu chegasse até aqui!!!.

Agradeço ao meu amor, ANA TERRA, pela compreensão e apoio nos momentos difíceis dessa jornada. Saiba que eu te amo muito meu amor e que sem você nada disso seria possível !!!!! Para terminar, gostaria de agradecer a Deus!!!!.

# RESUMO

O desenvolvimento da tecnologia tem permitido que empresas e organizações acumulem grandes quantidades de dados em bancos de dados. Porém, a posse destes dados não se traduz imediatamente em posse de informação útil. Para isso, são necessárias técnicas e ferramentas que explorem os dados de forma eficiente. O processo de descoberta de conhecimento em base de dados é usado para extrair informação útil, não trivial e previamente desconhecida de volumes de dados. Dentre as técnicas de mineração de dados, a mineração visual de dados é a mais intuitiva por usar a habilidade humana de rapidamente interpretar cenas visuais. Técnicas de mineração visual mapeiam dados em metáforas visuais a fim de permitir a interpretação dos dados. O presente trabalho desenvolveu uma ferramenta de mineração visual de dados que usa grafos como sua principal metáfora visual. Esta ferramenta, denominada GraphMiner mapeia dados estruturados, armazenados em bancos de dados relacionais, para grafos facilmente interpretáveis por seres humanos. Além disto, a GraphMiner provê uma série de controles de visualização e de consulta sobre o grafo produzido. Ela cria um ambiente que permite a exploração de dados relacionais de forma eficiente e intuitiva.

**Palavras-chave:** Mineração Visual de Dados, Grafos, Bancos de Dados relacionais.

# ABSTRACT

The development of technology has allowed companies and organizations to accumulate great amounts of data in databases. However, the ownership of these data is not expressed in ownership of useful information immediately. For this, techniques and tools are necessary that explore the data of efficient form. The process of knowledge discovery in database is used to extract useful, not trivial and previously unknown information from volumes of data. Amongst the techniques of mining of data, visual data mining is intuitive for using the ability human being quickly to interpret visual scenes. Techniques of visual mining map given in visual metaphors in order to allow the interpretation of the data. The present work developed a tool of visual mining of data that uses graphs as its main visual metaphor. This tool, called maps GraphMiner structuralized, stored data in relational databases, for easily interpretable graphs for human beings. Moreover, the GraphMiner to provide a series with consultation and visualization controls on the produced graph. It creates an environment that allows the exploration of relational data of efficient and intuitive form.

**Keywords:** Visual Data Mining, Graphs, Relational Databases

# LISTA DE FIGURAS

2.1	Modelo de Referência de Visualização [S. K. Card and Shneiderman, 1999].	6
2.2	Caracterização de dados, exemplos de domínios diferentes [Branco, 2003].	8
2.3	Etapas do processo de mineração de dados. . . . .	10
2.4	Ferramenta de Mineração Visual de Dados. . . . .	12
2.5	Visualização por Coordenadas Paralelas [Keim, 2002]. . . . .	13
2.6	Visualização em pixels: Técnica de segmentos circulares [Keim, 2002]. . .	14
2.7	(a) Ícone que representa cinco variáveis;(b) Família de ícones. . . . .	15
2.8	Imagem da Região dos Grandes Lagos com ícones de cinco dimensões [Ankerst, 2000]. . . . .	16
3.1	Metodologia de visualização de dados estruturados [Niggemann, 2001]. . .	20
3.2	Exemplo de visualização de grafos usando força-dirigida [Niggemann, 2001].	21
3.3	Visualização de grafos usando desenho orientado a nível [Niggemann, 2001].	23
3.4	Primeira etapa do algoritmo orientado a nível. . . . .	23
3.5	Visualização de grafo usando o IGD. . . . .	27
3.6	Visualização de grafo usando o GRAAPA. . . . .	27
3.7	Visualização de grafo usando o ZoomGraph. . . . .	28
3.8	Visualização de grafo resultante da busca atreves do Kartoo. . . . .	29
3.9	Visualização de grafo através do TouchGraph. . . . .	30
4.1	Relacionamento entre dados e metadados. . . . .	37
4.2	Exemplo de dados organizados em uma tabela. . . . .	38
4.3	(a) Relacionamentos entre as tabelas de um banco relacional. (b) Tabela que armazena as informações sobre os relacionamentos. . . . .	39
4.4	Passos para construção de um grafo a partir de um banco de dados relacional.	40
5.1	Arquitetura da ferramenta GraphMiner. . . . .	45
5.2	Diagrama das classes de acesso a dados. . . . .	47
5.3	Visualização de um DER pela GraphMiner. . . . .	48
5.4	Diagrama das Classes do TouchGraph adaptadas para a GraphMiner. . .	49
5.5	Grafo de dados visualizados na GraphMiner . . . . .	51
5.6	Árvore de dados que é usada para a criação dos controles de consulta. . .	52
5.7	Barra de intervalo usada como controle de consulta. . . . .	52
5.8	Diagrama das classes que implementam o controle por barras de intervalo.	53
5.9	Grafo antes e depois da utilização dos controles de consulta. . . . .	54
5.10	Grafo antes e depois da utilização dos controles de interação baseado em aproximação. . . . .	55
5.11	Barra de rolagem da classe GMPanel. . . . .	55
5.12	Diagrama das classes que implementam o controle interação com usuário zoom. . . . .	56
5.13	Grafo antes (a) e depois (b) da utilização dos controles de interação que permite girar o grafo em torno do nó centralizado (de cor verde). . . . .	57

5.14	Diagrama das classes que implementam o controle interação para rotação.	57
5.15	Grafo com nível de exibição configurado com valor maior que quatro. (b) Grafo com nível de exibição configurado com valor igual a um. . . . .	58
5.16	Diagrama das classes que implementam o controle de níveis de exibição do grafo. . . . .	59
5.17	Caixas de seleção dos atributos das entidades do banco para mapeamento dos atributos visuais do grafo. . . . .	59
5.18	Grafo com mapeamentos de atributos visuais. . . . .	61
5.19	Diagrama das classes que implementam o mapeamento dos atributos visuais do grafo. . . . .	61
5.20	Detalhes sobre demanda de um nó do grafo. . . . .	62
5.21	Diagrama das classes do controle de detalhes sob demanda. . . . .	62
5.22	(a) Grafo exibindo o rótulo completo. . . . .	63
5.23	(b) Grafo exibindo apenas o primeiro caractere. . . . .	64
6.1	Tabela de Pessoas do banco de dados de projetos do NUPERC. . . . .	69
6.2	Tabela de projetos usada no exemplo de uso do NUPERC. . . . .	70
6.3	Relação utilizada no exemplo de uso do NUPERC. . . . .	71
6.4	Grafo dos dados de projetos da base relacional do NUPERC. . . . .	72
6.5	Grafo com atributos visuais mapeados com atributos do banco. . . . .	73
6.6	Grafo com os nós de projeto filtrados pelo atributo valor. . . . .	74
6.7	Entidade “tipo” usado no exemplo de uso InfoVis. . . . .	75
6.8	Tabela “institutos de pesquisa” usada neste exemplo de uso. . . . .	76
6.9	Relação “Publicações” usada no exemplo de uso InfoVis. . . . .	76
6.10	Grafo dos dados de publicações do relatório InfoVis. . . . .	77
6.11	Grafo com cor do nó mapeada para um atributo numérico do banco. . . . .	78
6.12	Grafo com rótulo dos nós de publicações mapeados para o título. . . . .	78
6.13	Controles visuais com caixas de verificação (a) e botões de opção (b) [Almeida, 2003]. . . . .	80



# LISTA DE TABELAS

3.1	Resumo sobre os principais algoritmos abordados . . . . .	26
-----	---	----

# SUMÁRIO

<b>Capítulo 1—INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivo deste Trabalho . . . . .	2
1.2 Abordagem Adotada . . . . .	2
1.3 Organização da Dissertação . . . . .	3
<b>Capítulo 2—VISUALIZAÇÃO DE INFORMAÇÃO E MINERAÇÃO VISUAL DE DADOS</b>	<b>5</b>
2.1 Caracterização dos Dados . . . . .	7
2.2 Mineração de dados . . . . .	10
2.3 Mineração Visual de Dados . . . . .	11
2.4 Técnicas de visualização para Mineração Visual de Dados . . . . .	12
2.4.1 Projeções Geométricas . . . . .	12
2.4.2 Técnicas Orientadas a Pixels . . . . .	13
2.4.3 Técnicas Iconográficas . . . . .	15
2.4.4 Técnicas Hierárquicas . . . . .	16
2.5 Considerações Finais . . . . .	18
<b>Capítulo 3—VISUALIZAÇÃO DE INFORMAÇÃO BASEADA EM GRAFOS</b>	<b>19</b>
3.1 Técnicas de visualização de grafos . . . . .	21
3.1.1 Força-Dirigida (Force-Directed) . . . . .	21
3.1.2 Desenho Orientado a Nível (Level-Oriented Drawing) . . . . .	23
3.1.3 Outras Técnicas . . . . .	24
3.2 Classificação das Técnicas de Visualização . . . . .	25
3.3 Ferramentas para Visualização Baseada em Grafos . . . . .	26
3.3.1 IGD (Interactive Graph Drawing) . . . . .	26
3.3.2 GRAPPA . . . . .	27
3.3.3 Zoomgraph . . . . .	28
3.3.4 KARTOO . . . . .	29
3.3.5 TouchGraph . . . . .	29
3.3.6 Outras Ferramentas . . . . .	31
3.4 O Cenário Atual das Ferramentas de Mineração Visual Baseadas em Grafos	31
3.4.1 Áreas de Aplicação . . . . .	32
3.4.2 Algoritmos Utilizados . . . . .	33
3.4.3 Controles de visualização . . . . .	33
3.4.4 Entrada de dados . . . . .	34
3.5 Considerações Finais . . . . .	34

<b>Capítulo 4—O DESENHO DE GRAFOS A PARTIR DE BANCOS DE DADOS RELACIONAIS</b>	<b>35</b>
4.1 Bancos de dados e Metadados . . . . .	36
4.2 O Modelo Relacional . . . . .	37
4.3 O Mapeamento de Dados em Bancos Relacionais para Grafos . . . . .	39
4.4 Considerações Finais . . . . .	41
<b>Capítulo 5—A FERRAMENTA GRAPHMINER</b>	<b>42</b>
5.1 Requisitos Funcionais da Ferramenta . . . . .	43
5.2 Desenvolvimento da Ferramenta . . . . .	44
5.3 Arquitetura da Ferramenta . . . . .	44
5.3.1 Módulo de Entrada de Dados . . . . .	46
5.3.2 Módulo de Apresentação Visual . . . . .	48
5.3.3 Módulo de Controle de Consulta . . . . .	51
5.3.4 Módulo de Controle de Interação . . . . .	54
5.3.4.1 Controle de Aproximação (zoom) . . . . .	55
5.3.4.2 Controle de Rotação . . . . .	56
5.3.4.3 Controle do Raio de Exibição . . . . .	57
5.3.4.4 Controle de Mapeamentos . . . . .	58
5.3.4.5 Outros Controles . . . . .	62
5.4 Comparando a ferramenta GraphMiner com outras ferramentas . . . . .	64
5.5 Considerações Finais . . . . .	66
<b>Capítulo 6—EXEMPLOS DE USO DA FERRAMENTA GRAPHMINER</b>	<b>67</b>
6.1 Exemplo de uso NUPERC . . . . .	67
6.1.1 Caracterização da Organização . . . . .	67
6.1.2 Domínio de Aplicação . . . . .	68
6.1.3 Sobre os Dados . . . . .	69
6.1.4 Desenvolvimento do exemplo de uso . . . . .	69
6.2 Exemplo de uso InfoVis . . . . .	74
6.2.1 Caracterização do Relatório . . . . .	74
6.2.2 Domínio de Aplicação . . . . .	75
6.2.3 Desenvolvimento do Estudo de Caso . . . . .	75
6.3 Pontos Fortes e Fracos Observados na GraphMiner . . . . .	78
6.4 Considerações Finais . . . . .	81
<b>Capítulo 7—Conclusões</b>	<b>82</b>
7.1 Contribuições . . . . .	82
7.2 Trabalhos Futuros . . . . .	83

## INTRODUÇÃO

A quantidade de informações armazenadas em repositórios computacionais de dados tem crescido mais rápido que a nossa capacidade de explorá-la. Além disso, a complexidade de muitos conjuntos de dados supera a habilidade dos seres humanos de identificar os conceitos subjacentes às informações armazenadas.

Segundo [Keim, 2002], não há na história relatos da geração de dados em volumes elevados como nos níveis atuais, o que torna as tarefas de exploração e análise cada vez mais difíceis. Neste cenário, a exploração visual de dados surgiu como uma ferramenta útil às organizações que querem melhor explorar as informações contidas em seus repositórios.

O ser humano resolve o problema de compreender uma situação através da identificação e classificação de informação estruturada [Niggemann, 2001]. Em muitos casos, as estruturas classificadas podem ser visualizadas de maneira que o resultado dessa visualização facilite a compreensão de estruturas conhecidas e de padrões que se repetem.

A escolha da metáfora visual mais adequada para visualização de estruturas é fundamental para facilitar o entendimento humano sobre os dados. Nesse sentido, grafos são estruturas visuais interessantes, pois, destacam as relações implícitas em dados no formato estruturado. O problema de desenhar grafos na tela de um computador é tratado há algum tempo pela ciência da computação. Existem diversos algoritmos que tratam esse problema dentre os quais destaca-se a família de algoritmos denominada força-dirigida.

Um problema relevante é o de obter algo a mais que a simples representação

gráfica de um grafo. O objetivo neste caso é enriquecer os desenhos de modo tal que possam transmitir algo mais que o expressado pela topologia do grafo. Isto nos move ao campo da mineração visual de dados.

A mineração visual de dados permite combinar a capacidade de construir complexas cenas visuais com controles de seleção iterativa dos dados. Estas funcionalidades combinadas permitem que um especialista possa ele próprio, de forma iterativa, explorar os dados.

Hoje em dia, os mecanismos mais comuns para o armazenamento de dados estruturados são bancos de dados relacionais. Esses mecanismos, além de armazenar os dados estruturados, tratam de forma semelhante os metadados que descrevem esses dados. Isto facilita a obtenção das relações entre os dados que são elementos essenciais para mineração visual através de grafos.

## **1.1 OBJETIVO DESTE TRABALHO**

O objetivo deste trabalho foi a implementação de uma ferramenta que apóie a descoberta de informações úteis em bancos de dados relacionais através da representação gráfica desses dados usando grafos como metáfora visual. A principal contribuição deste trabalho é oferecer um mecanismo que permite a exploração de dados relacionais de forma simples e eficiente.

## **1.2 ABORDAGEM ADOTADA**

Para atingir o objetivo proposto, o estado da arte na área de mineração visual de dados, em geral, e visualização de grafos, em particular, foram estudados. Os requisitos para uma ferramenta de mineração visual de dados foram especificados, e uma abordagem para alimentar esta ferramenta com dados armazenados em bancos de dados relacionais foi definida. A ferramenta, denominada GraphMiner, foi então implementada. Esta

ferramenta é genérica o suficiente para que seja independente de um banco de dados ou domínios específicos. Para tanto, foi realizado um estudo sobre ferramentas e modelos existentes e, no processo de desenvolvimento da ferramenta, foram adotadas técnicas de programação orientada a objetos já consolidadas na área de engenharia de software. Além disso, dois exemplos de uso foram desenvolvidos, com o intuito de subsidiar a avaliação da usabilidade da ferramenta.

### **1.3 ORGANIZAÇÃO DA DISSERTAÇÃO**

A estrutura deste documento foi definida com o objetivo de fornecer ao leitor subsídios para o fácil entendimento do trabalho realizado. Após a introdução são apresentados três capítulos contendo a fundamentação teórica e revisão da literatura relacionada ao tema. A descrição do trabalho desenvolvido nesta dissertação encontra-se nos próximos dois capítulos. Após a conclusão são apresentadas as referências bibliográficas utilizadas neste trabalho. Sendo assim, esta dissertação está dividida nos seguintes capítulos:

- i) Capítulo 1 - descreve a motivação, objetivos e escopo da pesquisa, como também a organização do documento da dissertação.
- ii) Capítulo 2 - apresenta a fundamentação teórica e a revisão de literatura sobre o tema Visualização de Informação.
- iii) Capítulo 3 - apresenta as linhas teóricas existentes e o estado da arte de visualização de informações baseadas em grafos, no intuito de contextualizar os objetivos da dissertação.
- iv) Capítulo 4 - apresenta a construção visual de um grafo na tela de um computador a partir de dados armazenados em bancos de dados relacionais.
- v) Capítulo 5 - este capítulo descreve detalhadamente a ferramenta GraphMiner e sua implementação.

- vi) Capítulo 6 - neste capítulo é feita uma avaliação da usabilidade da ferramenta através de dois exemplos de uso.
- vii) Capítulo 7 - apresenta as conclusões deste trabalho e comenta a evolução da ferramenta, relacionando os trabalhos futuros que podem ser realizados.

# VISUALIZAÇÃO DE INFORMAÇÃO E MINERAÇÃO VISUAL DE DADOS

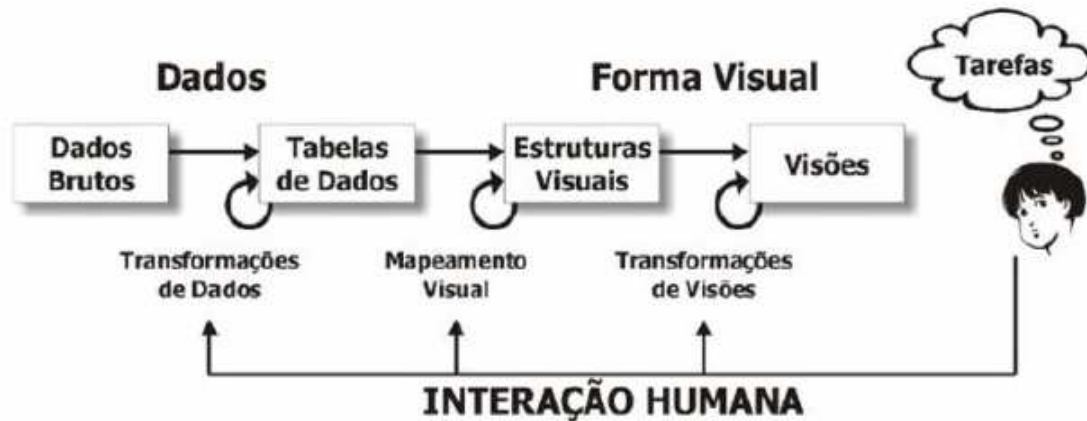
A visualização é uma disciplina que reúne métodos permitindo a interação com representações visuais de dados, exibidas como imagens, para ganhar conhecimento sobre os dados e os fenômenos que eles representam. O propósito da visualização é a percepção com o objetivo de descoberta e tomada de decisões [S. K. Card and Shneiderman, 1999]. O procedimento de exploração visual de dados pode ser visto ainda, como um processo de geração de hipóteses em que a visualização garante ao usuário entendimento sobre os dados permitindo a criação de suas suposições [Keim, 2002].

A combinação de aspectos como flexibilidade, criatividade e conhecimento geral do homem aliados à enorme capacidade computacional e de armazenamento dos computadores, são pontos decisivos para o sucesso do processo de descoberta de conhecimento a partir de volume de dados. A exploração visual visa integrar o ser humano no processo da exploração, aplicando sua habilidade de percepção aos conjuntos de dados disponíveis nos sistemas computacionais atuais [Keim, 2002].

As técnicas de mineração visual têm alto valor na análise exploratória de dados e na prospecção de grandes repositórios. A exploração visual é especialmente útil quando pouco é sabido sobre os dados e quando os seus objetivos são vagos. As etapas essenciais a serem consideradas no uso de determinada técnica de visualização, ou no desenvolvimento de novas técnicas, podem ser identificadas por meio de modelos de referência de visualização. Uma descrição de visualização como uma seqüência de mapeamentos ajustáveis de dados para uma representação visual é dada por [S. K. Card and Shneiderman, 1999] e ilustrada na Figura 2.1. Ela possibilita a interação do usuário com o espaço de in-



formação, a fim de alcançar o que foi chamado de cristalização do conhecimento.



**Figura 2.1.** Modelo de Referência de Visualização [S. K. Card and Shneiderman, 1999].

Diversas áreas de estudo humanas estão interessadas na criação de metáforas visuais, e muitas foram beneficiadas com a produção de cenas voltadas à visualização. A área científica, por exemplo, vem se beneficiando da capacidade computacional e das visualizações complexas alcançadas, na chamada Visualização Científica [Branco, 2003]. Este tipo de visualização utiliza dados de fenômenos naturais do mundo físico. O computador é usado para tornar visível alguma propriedade de objetos de interesse. Por exemplo, para representar o volume e a composição da água em um reservatório pode-se fazer uso de abstrações, mas elas são baseadas num espaço físico.

Um outro aspecto a ser ressaltado é que há uma grande quantidade de informação não-física no mundo contemporâneo que também demandam meios que viabilizem extração de conhecimento a partir desta informação. A visualização de informações não-físicas é simplesmente chamada de visualização de informação. Este tipo de visualização tem uma diferença chave da visualização científica. A diferença está na inexistência de um mapeamento espacial claro, o que dificulta tornar visíveis as propriedades dos objetos de interesse. A questão é como mapear abstrações não-espaciais em uma representação visual efetiva. O esforço de estender a visualização para a área da abstração de fenômenos não-físicos é o que nos move ao campo da visualização de informação, e tem sido motivado não só pelo grande acúmulo de informações na atualidade, mas também

pela complexidade intrínseca ao processo de se alcançar um mapeamento visual eficaz.

A visualização de informação origina-se no emprego de representações visuais para exibir dados abstratos em diversas comunidades de pesquisa, e só mais tarde percebe-se a convergência para o uso do poder computacional nas representações visuais. Por sua vez, a visualização científica tem sua origem notoriamente ligada às ciências que abordam o espaço físico. A distinção entre estas vertentes da visualização é aceita por grande parte dos autores encontrados na literatura.

Outra questão quanto a função da visualização na obtenção de conhecimento, refere-se ao fato dela fornecer uma faixa maior de elementos mais facilmente identificáveis, em relação aos outros sentidos de percepção. Embora a representação de dados abstratos possa ser realizada por meio de sistemas que abordem outras formas de percepção, é um padrão de fato que a visualização seja apontada como um ponto inicial neste esforço [S. K. Card and Shneiderman, 1999] [Shneiderman, 1996].

## 2.1 CARACTERIZAÇÃO DOS DADOS

No modelo de referência de Visualização da Figura 2.1, o objetivo central, que é mapear visualmente tabelas de dados em uma metáfora visual, serve como fase intermediária entre as etapas que envolvem tratamento de dados e as etapas que lidam com a forma visual. As tabelas de dados baseiam-se em relações matemáticas, enquanto as estruturas visuais são baseadas nas propriedades gráficas processadas pela visão humana. A terminologia encontrada na literatura, no que se refere a dados, não é consistente, já que vem sendo criada por muitas disciplinas como estatística, engenharia, e ciência da computação, entre outras [Wong and Bergeron, 1997].

Em um processo de visualização, a determinação de qual técnica deve ser empregada em uma determinada aplicação ou situação merece bastante atenção. Uma caracterização dos dados seria a consideração inicial na escolha de uma técnica de visualização

[Branco, 2003]. Na tentativa de enquadramento de aplicações em técnicas, alguns autores propõem certas classificações. SHNEIDERMAN [Shneiderman, 1996], por exemplo, classificou as técnicas segundo os tipos de dados e as tarefas a serem realizadas pelo usuário. Segundo ele, os dados podem ser: temporais, unidimensionais (1D), bidimensionais (2D), tridimensionais (3D), multidimensionais (nD), dirigidas à visualização de hierarquias e de relacionamentos (grafos). FREITAS e WAGNER [Freitas and Wagner, 1995] apresentam uma proposta de caracterização de dados baseada em critérios como: classe (tipo) de informação, tipos de valores, e natureza e dimensão do domínio. A Figura 2.2, exibida a seguir, foi feita por [Branco, 2003]. Ela reúne classificações propostas com exemplos em diferentes domínios.

Critério	Classe <sup>a</sup>	Exemplo
Classe de Informação	Categoria	Gênero
	Escalar	Temperatura
	Vetorial	Grandezas físicas associadas à dinâmica de fluidos
	Tensorial	Link num hiperdocumento
Tipo de Valores	Relacionamento	Link num hiperdocumento
	Alfanumérico	Gênero
	Númérico (inteiro, real)	Temperatura
Natureza do Domínio	Simbólico	Link num hiperdocumento
	Discreto	Marcas de automóveis
	Contínuo	Superfície de um terreno
Dimensão do Domínio	Contínuo-discretizado	Anos (tempo discretizado)
	1D	Fenômeno ocorrendo no tempo
	2D	Superfície de um terreno
	3D	Volume de dados médicos
	n-D	Dados de uma população

**Figura 2.2.** Caracterização de dados, exemplos de domínios diferentes [Branco, 2003].

De forma geral, os valores assumidos por variáveis podem ser classificados nos formatos básico nominal e quantitativo. O primeiro apresenta valores claramente distintos, discretos e enumeráveis. O segundo representa valores numéricos, contínuos, sobre os quais podem ser aplicadas operações aritméticas. Os dados nominais podem ser categóricos, em que os valores não têm uma relação de ordem (ex: “verde”, “vermelho”, “azul”, “GM”, “Fiat”, “Ford”), ou ordinais, apresentando relação de ordem (ex:

“segunda”, “terça”, “quarta”, “básico”, “intermediário”, “avançado”). Por sua vez, os dados quantitativos podem ser intervalos, nos quais os valores são ordenados e medidos em unidades fixas e iguais (ex: ano), podem ser uma razão, em que os valores são ordenados em um escala de medidas na qual é definido inerentemente um valor de referência zero (ex: distâncias) e alguns sub-tipos particulares, tais como datas, horas e coordenadas espaciais.

Esta caracterização mais genérica, torna mais compreensível como podem ocorrer as transformações nos dados apontadas no modelo de referência da Figura 2.1. Por exemplo, uma variável originalmente quantitativa, tal como o tempo de duração de um filme  $[0, 360]$ , pode ter seus valores separados em faixas como (“curto”, “médio”, “longo”), denotando agora um tipo de variável nominal / ordinal.

KEIM [Keim, 2002] faz uma caracterização de acordo com o número de variáveis (dimensionalidade) e complexidade relacionadas ao conjunto de dados que é mais abrangente que a proposta por [Shneiderman, 1996]. Esses conjuntos são classificados como unidimensional, bidimensional, multidimensional e ainda:

- i) Texto / Hipertexto: Nem todos os tipos de dados podem ser descritos nos termos de dimensionalidade. Em tempos de WWW, tipos de dados importantes são: o texto, o hipertexto e o conteúdo multimídia de páginas web. Estes tipos não podem ser facilmente descritos por números e conseqüentemente, a maioria dos padrões de visualização não podem ser aplicados. Na maioria dos casos, primeiramente uma transformação é necessária antes que as técnicas possam ser usadas sobre estes dados.
- ii) Grafos / Hierarquias: Os registros de dados têm freqüentemente algum relacionamento com outras partes da informação. Grafos são muito usados para representar tais interdependências. Um grafo consiste em um conjunto de objetos, chamados de nós, e conexões entre eles, chamadas de arestas [Keim, 2002]. Alguns exemplos

são as inter-relações de e-mail entre as pessoas, seu comportamento em relação as compras, a estrutura de arquivos do disco rígido e os links Web. A visualização de dados baseada em grafos será melhor discutida nas seções subseqüentes.

- iii) Algoritmos / Software: Lidar com grandes projetos de software é um desafio. O objetivo da visualização de software é dar suporte ao desenvolvimento de software ajudando compreender algoritmos, como por exemplo mostrando o fluxo da informação em um programa, para facilitar a compreensão do código escrito ou ainda, representando a estrutura das milhares de linhas de código fonte de forma gráfica para que o programador possa eliminar erros do código [B. A. Price and Small, 1998].

## 2.2 MINERAÇÃO DE DADOS

Em MENDONCA [Mendonça, 2001] encontra-se uma definição de mineração de dados abrangente que cobre uma grande quantidade de métodos, técnicas e ferramentas: “mineração de dados é o processo de obtenção de informação que seja nova, não trivial, e útil de bases de dados”. O processo de mineração de dados compreende quatro etapas: seleção, pré-processamento, mineração e assimilação.



**Figura 2.3.** Etapas do processo de mineração de dados.

A Figura 2.3 exibe as etapas do processo de mineração onde há inicialmente, a escolha (seleção) dos dados a serem explorados. Na segunda etapa, o pré-processamento, os dados sofrerão um processo de formatação, limpeza e adaptação, a fim de que possam ser trabalhados por um algoritmo de mineração. Na etapa seguinte, os dados são miner-

ados de forma propriamente dita, permitindo a descoberta de informação desconhecida até então. Na fase de assimilação essa informação é então interpretada e transformada em conhecimento e ações por um indivíduo ou por uma organização.

### 2.3 MINERAÇÃO VISUAL DE DADOS

Muitas das ferramentas modernas de visualização, combinam a capacidade de construir complexas cenas visuais com controles de seleção iterativa dos dados. Estas funcionalidades combinadas permitem que um perito possa ele próprio iterativamente explorar os dados. Este tipo de exploração interativa de dado é chamada de mineração visual de dados [Mendonça, 2001]. Segundo [Almeida, 2003], uma ferramenta para visualização de dados normalmente respeita alguns princípios funcionais básicos:

- i) Faz o uso de atributos visuais como forma, cor, posicionamento, tamanho para produzir cenas visuais facilmente interpretáveis por um ser humano.
- ii) Possibilita a navegação interativa na tela bidimensional de um computador, possibilitando zoom, rotação, re-posicionamento, e varreduras sobre a área exibida.
- iii) Permite o controle interativo dos formatos de apresentação e dos atributos visuais exibidos na metáfora visual.
- iv) Permite o detalhamento de qualquer item específico de dado.

A Figura 2.4 mostra uma ferramenta para mineração visual de dados no formato hierárquico. A área de visualização da ferramenta é mostrada no lado esquerdo da figura usando retângulos. Os atributos visuais cor, tamanho e rótulos dos retângulos são associados com atributos reais dos dados produzindo a visualização. As características das estruturas visuais do modelo como o ajuste do tamanho dos retângulos, das cores que representarão os atributos e a formação dos agrupamentos com a escolha da hierarquia dos dados é feita através dos controles de interação presentes no lado inferior direito da figura. O canto superior direito da figura mostra os detalhes, obtidos sob demanda, de um item específico de dados.

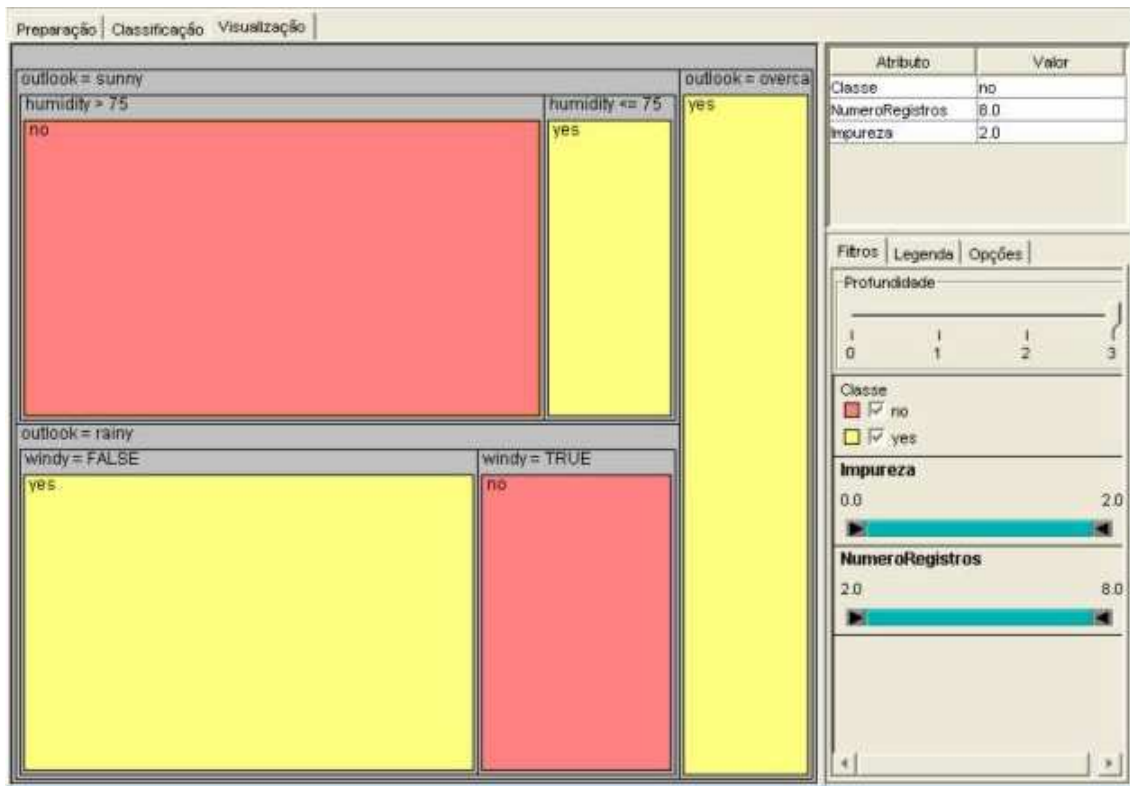


Figura 2.4. Ferramenta de Mineração Visual de Dados.

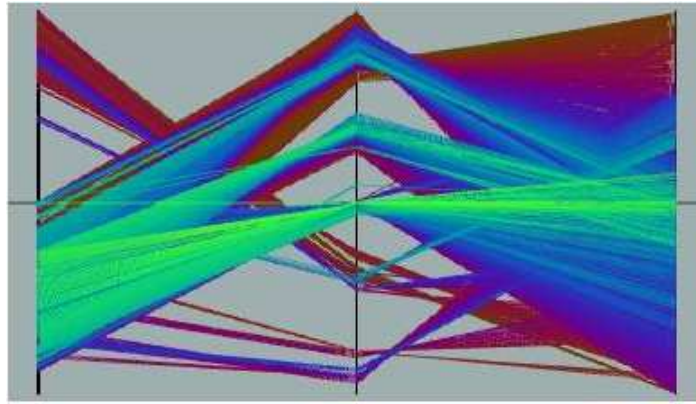
## 2.4 TÉCNICAS DE VISUALIZAÇÃO PARA MINERAÇÃO VISUAL DE DADOS

[Keim and Kriegel, 1996] e [Wong and Bergeron, 1997] enumeram técnicas bem conhecidas para visualização de conjuntos de dados multidimensionais e as descrevem segundo critérios diferenciados, ainda que com alguns elementos em comum. Ainda de acordo com [Keim and Kriegel, 1996], os autores agruparam as técnicas de exploração visual de dados em seis categorias, segundo a abordagem adotada para o mapeamento dos dados em uma representação visual, são elas: Projeções Geométricas, Orientadas a Pixels, Iconográficas, Hierárquicas, Híbridas e Baseadas em Grafos.

### 2.4.1 Projeções Geométricas

A técnica de Projeções Geométricas tem como objetivo principal encontrar transformações de interesse em conjuntos de dados multidimensionais [Keim, 2002]. Uma técnica muito utilizada, chamada de Coordenadas Paralelas (Parallel Coordinates), foi apresentada por [Inselberg and Dimsdale, 1990] como uma tecnologia de Geometria Com-

putacional, posteriormente contextualizada em visualização de informação [Wegman, 1990], e destaca-se pela perspectiva multidimensional conferida à representação visual. Nela, um espaço de dimensão  $k$  é mapeado para um espaço visual bidimensional, usando  $k$  eixos eqüidistantes e paralelos a um dos eixos principais ( $x$  ou  $y$ ). Cada um dos  $k$  eixos representa uma dimensão do conjunto de dados, sobre o qual é mapeado linearmente, do menor ao maior, o intervalo de valores de dados correspondente. Cada item de dado é exibido como uma linha que intercepta cada eixo no ponto correspondente ao valor do atributo associado ao eixo.



**Figura 2.5.** Visualização por Coordenadas Paralelas [Keim, 2002].

A Figura 2.5 exibe o esquema de utilização de coordenadas paralelas para visualização de dados. Embora simples, a técnica mostra-se poderosa para identificar diferentes distribuições de dados e dependência funcional entre atributos. Em contrapartida, apresenta problemas como a sobreposição de linhas para grandes volumes de dados e, por conseguinte, uma baixa quantidade de itens de dados que podem ser apresentados simultaneamente sem a ocorrência de congestionamento visual, cerca de 1.000 itens, segundo KEIM [Keim and Kriegel, 1996].

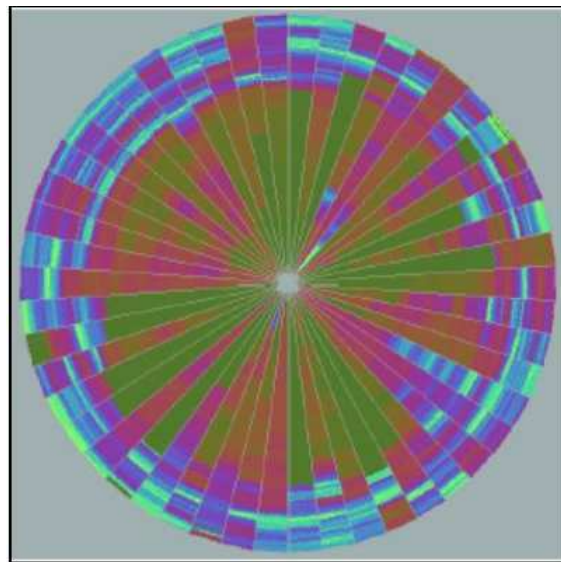
#### 2.4.2 Técnicas Orientadas a Pixels

Técnicas baseadas em pixels têm o objetivo de usar um pixel para representar cada valor de atributo, colorindo-o conforme um mapa de cores previamente fixado de acordo com a faixa de possíveis valores. Cada um destes atributos tem sua representação visual exibida em sub-janelas individuais na visualização [D. A. Keim and Seidl, 1994b]



[Keim and Kriegel, 1996]. Para conjuntos de dados com  $m$  atributos, a tela é dividida em  $m$  janelas. Correlação e dependência funcional podem ser detectadas pela análise de regiões correspondentes nas múltiplas janelas [Keim, 2000]. Uma das vantagens deste tipo de técnica é a grande quantidade de informação que pode ser exibida simultaneamente. Se um único atributo for apresentado em uma janela com resolução de  $1280 \times 1024$ , é possível exibir mais de um milhão de registros simultaneamente.

O sucesso na utilização desse tipo de técnica depende de alguns aspectos. Segundo KEIM [D. A. Keim and Seidl, 1994b] [Keim and Kriegel, 1996] [Keim, 2002], visto que cada valor de atributo é representado por um pixel, uma questão fundamental é a forma de mapear um conjunto de dados unidimensionais (um atributo) nas duas dimensões da tela. Esse mapeamento deve ser executado de maneira a fornecer boas propriedades de agrupamento (clustering), e manter algum significado semântico, favorecendo a percepção de relações existentes entre os dados. Assim, as técnicas orientadas a pixel recebem diferentes arranjos segundo diferentes propósitos e condições. A Figura 2.6 mostra a visualização de dados usando uma técnica baseada em pixels chamada de técnica de segmentos circulares. A idéia básica dessa abordagem é colorir cada dimensão com uma cor e agrupar os pixels pertencentes a essas dimensões em áreas adjacentes.

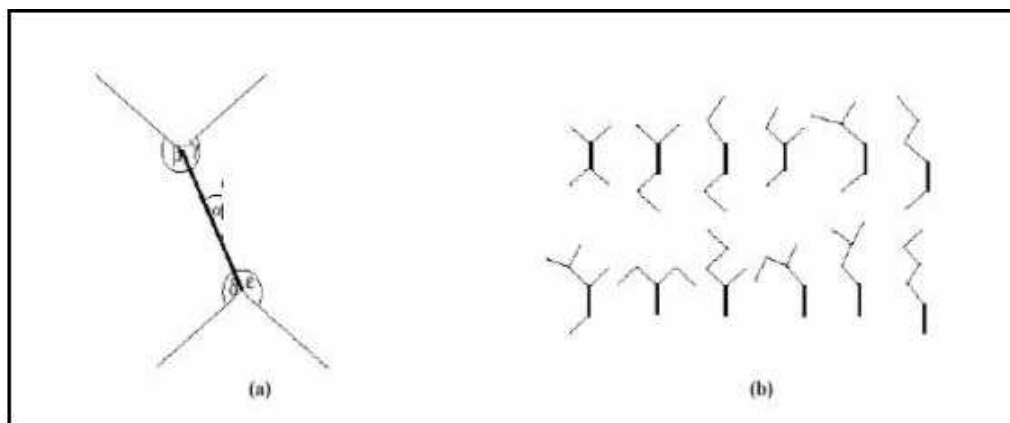


**Figura 2.6.** Visualização em pixels: Técnica de segmentos circulares [Keim, 2002].

### 2.4.3 Técnicas Iconográficas

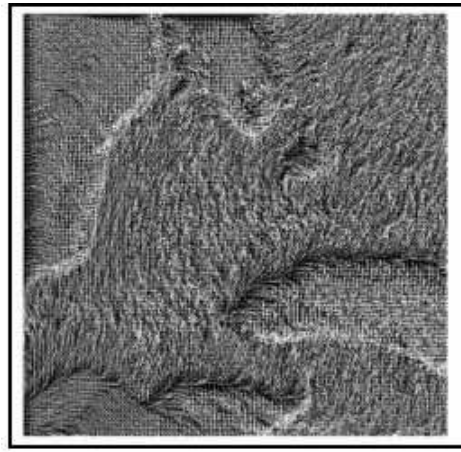
Uma outra classe de técnicas de exploração visual de dados é a iconográfica. A idéia é mapear os valores de um atributo de dados multidimensionais com as características de um ícone. Os ícones podem ser definidos de várias formas, alguns exemplos são: rostos pequenos [Cherno, 1973], ícones estrela [Ward, 1994] e ícones de varetas [Pickett and Grinstein, 1988].

Segundo PICKETT e GRINSTEIN [Pickett and Grinstein, 1988], a técnica de varetas utiliza a tela bidimensional para mapear duas dimensões dos dados e as demais são mapeadas para os ângulos e/ou comprimentos de segmentos de um ícone formado por múltiplos segmentos de reta. A Figura 2.7 (a) exibe um ícone com uma configuração que apresenta cinco variáveis, na qual uma dimensão é mapeada pela inclinação do corpo do ícone, e as orientações das varetas permitem mapear outras quatro dimensões.



**Figura 2.7.** (a) Ícone que representa cinco variáveis;(b) Família de ícones.

Uma família de varetas é ilustrada na Figura 2.7 (b), em que cada uma tem um corpo e 4 segmentos. Quando mapeados na tela, os ícones (um para cada item de dado) formam texturas que variam de acordo com as características dos dados, permitindo a exploração da capacidade humana de perceber e distinguir texturas em imagens complexas [Wong and Bergeron, 1997], ao passo que os padrões percebidos na imagem podem, então, indicar dependência funcional entre os atributos visualizados [Keim and Kriegel, 1996] [Wong and Bergeron, 1997].



**Figura 2.8.** Imagem da Região dos Grandes Lagos com ícones de cinco dimensões [Ankerst, 2000].

A Figura 2.8 exibe a imagem formada por ícones que representam cinco variáveis, exemplificando como estas texturas podem ser formadas. Técnicas iconográficas conseguem mostrar grande quantidade de itens de dados, mas há limitações quanto à quantidade de dimensões que podem ser mapeadas sem que haja diminuição da capacidade de representar características perceptíveis dos dados. Outra questão é que o reconhecimento de um importante padrão na imagem depende da seleção de um mapeamento adequado dos parâmetros dos dados para os parâmetros visuais. O número destes possíveis mapeamentos visuais cresce muito em relação ao número de dimensões mapeadas, podendo tornar-se um fator limitador na visualização [Wong and Bergeron, 1997].

#### 2.4.4 Técnicas Hierárquicas

Técnicas hierárquicas de visualização fazem a divisão de um espaço multi-dimensional dos dados (não necessariamente hierárquicos) em subespaços organizados e exibidos na forma de hierarquia, projetando-os uns dentro dos outros [Branco, 2003]. Dentre as técnicas hierárquicas destaca-se o Empilhamento Dimensional (Dimensional Stacking), onde o espaço  $n$ -dimensional (discreto) é sub-dividido em espaços bidimensionais. Uma das maiores vantagens desta técnica é que ela não precisa de funções ou regras extras para que se possam desenhar os dados na representação [Keim and Kriegel, 1996] [Wong and Bergeron, 1997].

O uso desta técnica [Hoffman and Grinstein, 1999] é particularmente interessante na detecção de agrupamentos, de pontos com comportamentos discrepantes, e de padrões. Todavia, há limitações referentes ao número de atributos a serem visualizadas (acima de nove torna-se bastante difícil) e ao número de valores distintos que um mesmo atributo pode assumir. Além disso, a eficácia do processo de análise depende diretamente do arranjo hierárquico dos atributos (normalmente, os atributos mais importantes devem estar posicionados mais externamente) e dos critérios de categorização dos dados [Keim and Kriegel, 1996] [Wong and Bergeron, 1997] [Ward, 1994]. Esta última restrição é uma limitação artificial imposta pelo dispositivo gráfico, que pode ser minimizada pelo uso de técnicas de interação adequadas.

As representações tradicionais de hierarquias exibem a relação entre dois nós de uma árvore usando uma linha. Este tipo de abordagem tem dois problemas significantes: (1) Se Gasta muito espaço para arrumar os nós na tela de um computador e (2) estruturas hierárquicas não triviais geram árvores de visualização muito complexa. Na seção 2.3 a Figura 2.4 mostra a visualização de informações através de uma técnica chamada de Treemaps. Treemaps é uma técnica proposta por SHNEIDERMAN [Shneiderman, 1996] para visualização de estruturas hierárquicas. Ela é especialmente útil quando existem variáveis quantitativas nos dados. Esta técnica utiliza 100 por cento do espaço disponível para visualização das informações, mapeando a hierarquia em regiões retangulares. Ela é também bastante eficiente em mostrar os atributos dos nós a partir da codificação de tamanho e cores. O atributo visual "tamanho" é especialmente útil na representação de variáveis que podem ser decompostas hierarquicamente. Como mostrado na Figura 2.4, ele pode ser usado para fazer com que os nós que contenham informações de maior importância sejam colocados em regiões maiores que aqueles de menor importância. Isto permite aos usuários comparar os tamanhos dos nós e das sub-árvores, ajudando a mostrar padrões hierárquicos incomuns.

## 2.5 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas a fundamentação teórica e a revisão da literatura sobre o tema Visualização de Informação. Termos e conceitos importantes para contextualizar o trabalho aqui apresentado foram definidos.

Uma forma comum de organização de dados é o formato estruturado. Dados no formato estruturado são ideais para visualização usando grafos pois, os dados podem ser representados por nós e as relações inerentes aos dados podem ser representadas por arestas. No próximo capítulo serão apresentadas questões relacionadas à visualização de dados baseada em grafos, suas principais abordagens, assim como as mais recentes pesquisas e trabalhos realizados na área.

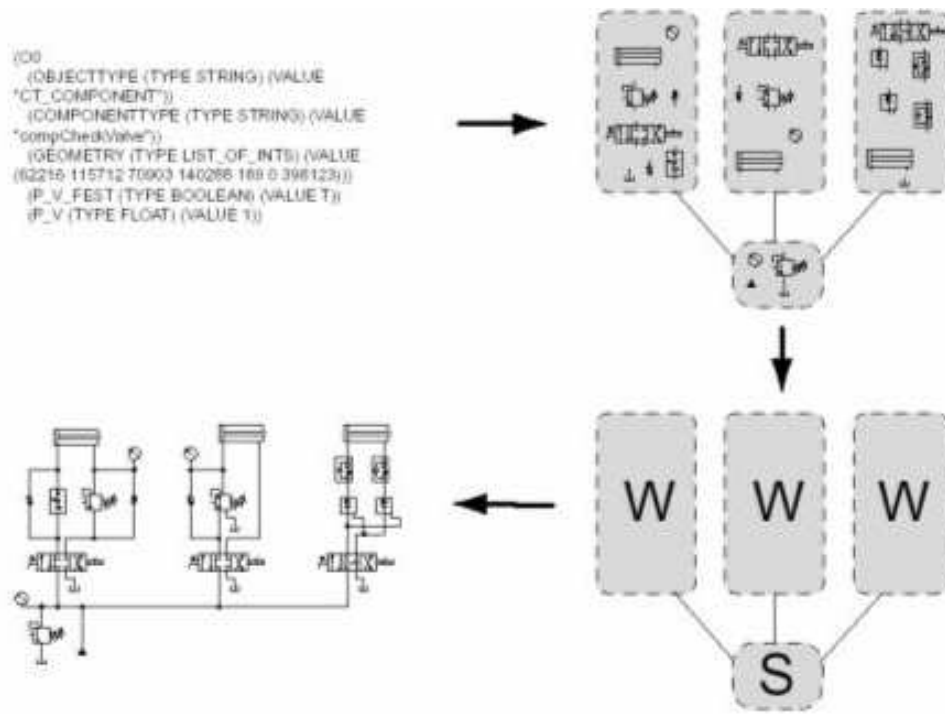
# VISUALIZAÇÃO DE INFORMAÇÃO BASEADA EM GRAFOS

A compreensão humana é baseada na identificação de estruturas. O problema de compreender uma situação complexa é resolvido pela mente humana através de uma abstração, que significa identificação e a classificação de informação estruturada. Um exemplo disso é o reconhecimento visual, que se baseia na identificação de objetos já conhecidos, ou ainda, o diagnóstico de defeitos em um dispositivo qualquer, que consiste na identificação de mau funcionamento em suas subestruturas [Niggemann, 2001].

Em muitos casos, as estruturas classificadas devem ser visualizadas de maneira que o resultado dessa visualização seja determinístico, isto é, as estruturas similares devem resultar em visualizações parecidas. Isto facilita a compreensão de estruturas conhecidas e de padrões que se repetem. Segundo NIGGEMANN [Niggemann, 2001], essas idéias conduzem à nova metodologia de visualização de dados no formato estruturado:

- i) Identificação da Estrutura: A primeira etapa identifica a estrutura dos modelos que é definida pelos submodelos e por seus relacionamentos.
- ii) Classificação da Estrutura: A segunda etapa consiste em classificar as subestruturas individuais.
- iii) Visualização da Estrutura: As subestruturas e seus relacionamentos são visualizados na terceira etapa. As subestruturas similares são visualizadas de maneira similar e as repetitivas são enfatizadas. A visualização coloca todas as subestruturas em um só plano.

A Figura 3.1 mostra um exemplo de visualização de um circuito hidráulico baseada na metodologia descrita. Primeiro, a estrutura do circuito textualmente definida



**Figura 3.1.** Metodologia de visualização de dados estruturados [Niggemann, 2001].

é identificada, isto é as sub-estruturas são esboçadas. Estas subestruturas são então classificadas: **S** denota uma subestrutura do tipo fonte e **W** significa uma subestrutura do tipo trabalho. Finalmente, o modelo visual é extraído. A disposição dá ênfase à informação estruturada colocando elementos da mesma subestrutura próximos. O padrão repetitivo das três unidades trabalhando é enfatizado pelo seu posicionamento paralelo.

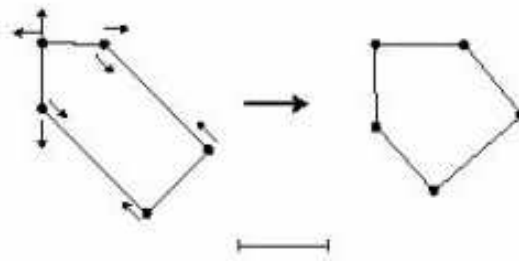
A escolha da metáfora visual adequada para visualização de estruturas é fundamental para facilitar o entendimento humano sobre os dados. Nesse sentido, grafos são estruturas visuais interessantes, pois, destacam as relações implícitas, presentes em dados no formato estruturado. Os dados, segundo a sua estruturação podem ser classificados em: não-estruturados, semi-estruturados e estruturados. Páginas da Web são exemplos de dados “semi-estruturados”, dados contidos em bancos de dados relacionais são ditos “estruturados”. Um exemplo de dado “não-estruturado” é um texto simples. Na seção 3.1 serão abordadas as principais técnicas de visualização de informações que usam grafos.

### 3.1 TÉCNICAS DE VISUALIZAÇÃO DE GRAFOS

O desenho de grafos não é uma área nova dentro da informática. A referência mais antiga encontrada foi escrita por KNUTH (1963). Alguns estudos sobre os mais recentes algoritmos dessa área foram feitos por BATTISTA [G. D. Battista and Tollis, 1994] [G. D. Battista and Tollis, 1999]. A maior parte desses algoritmos podem ser enquadrados em duas famílias: Força-Dirigida e Desenho Orientado a Nível.

#### 3.1.1 Força-Dirigida (Force-Directed)

Os métodos de força-dirigida usam o tamanho das arestas para desenhar um grafo de forma que a disposição resultante seja intuitiva ao ser humano. Os diferentes comprimentos, atribuídos inicialmente de forma aleatória, de cada uma das arestas são corrigidos movendo os nós correspondentes para uma posição melhor. Isto é ilustrado usando-se uma analogia do campo da física: Os nós correspondem a pequenos corpos, enquanto as arestas corresponderem às molas que os conectam. Partindo de uma disposição aleatória, as forças da mola movem os corpos em tais posições que deixam as molas o mais próximo do equilíbrio. No intuito de simular tal sistema em um computador, todas as forças que afetam um nó devem ser calculadas e interativamente aplicadas aos corpos. A Figura 3.2 descreve um exemplo da disposição de um grafo usando o algoritmo de força dirigida. As setas descrevem as forças nos nós e os movem para as posições mostradas no lado direito da figura.



**Figura 3.2.** Exemplo de visualização de grafos usando força-dirigida [Niggemann, 2001].

Uma alternativa a esta versão básica do algoritmo usa funções nas quais a



soma dos desvios entre os comprimentos ótimo e real de uma aresta é chamada energia. Os métodos de força-dirigida tentam encontrar uma disposição para minimizar essa energia. Assim, calcular forças sobre os nós e movê-los pode ser visto como uma busca para encontrar uma posição na qual o valor da função de energia seja mínimo.

A maior vantagem dos métodos de força-dirigida é o apelo intuitivo dos desenhos resultantes, que normalmente assemelham-se à noção estética humana de uma boa visualização de um grafo. Porém, algumas desvantagens também são observadas:

i) O fato desse tipo algoritmo modificar as posições dos elementos de um grafo durante a sua execução, restringe a sua utilização a grafos pequenos da ordem de centenas de nós [S. Guha and Shim, 1998].

ii) A cada linha de execução

$$O(n^2)$$

pares de nós são avaliados. Para a maioria dos algoritmos,  $O(n)$  passos são necessários para alcançar uma boa disposição resultando em uma complexidade de

$$O(n^3)$$

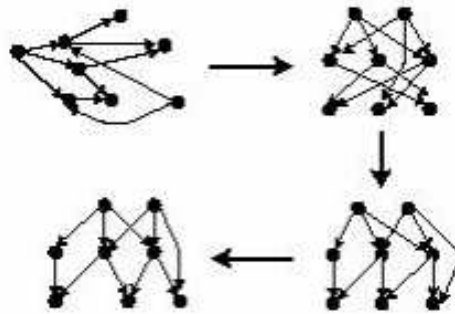
para os procedimentos de força-dirigida. HADANY [Hadany and Harel, 1999] apresenta propostas baseadas em aglomeração para reduzir a complexidade desses algoritmos.

Outras questões importantes são o cruzamento de arestas e outras restrições relativas a um bom desenho de um grafo. Essas questões são levadas em conta nas abordagens propostas por EADES [Eades, 1984] e BRANKE [J. Branke and Schmeck, 1997].

Uma técnica similar à força-dirigida é o MDS (*MultiDimensional Scaling*). Esta abordagem é mais voltada ao campo da estatística e da análise descritiva de dados [Togerson, 1952].

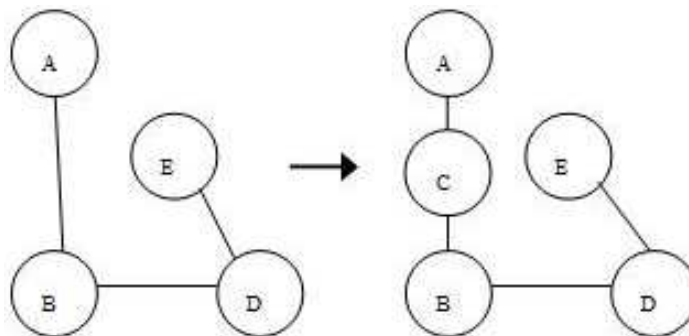
### 3.1.2 Desenho Orientado a Nível (Level-Oriented Drawing)

Os algoritmos de desenho de grafos orientados a nível são normalmente usados em grafos dirigidos e compreendem três passos [Sugiyama and Toda, 1981]: Primeiro os nós são distribuídos em camadas horizontais; depois são organizados de forma a minimizar o número de cruzamentos de arestas; e finalmente, a coordenada horizontal (X) é ajustada para cada um dos nós. A Figura 3.3 ilustra este processo.



**Figura 3.3.** Visualização de grafos usando desenho orientado a nível [Niggemann, 2001].

A primeira etapa associa nós com as camadas de tal maneira que os nós dentro de uma mesma camada não são conectados. Se os nós conectados não forem de camadas diretamente vizinhas, as arestas têm que cruzar diversas camadas. Para melhorar a aparência do desenho, os chamados “nós auxiliares” são introduzidos. A Figura 3.4 exemplifica a primeira etapa do algoritmo. Para uma aresta entre um nó a na camada 1 e um nó b na camada 3, um nó modelo c na camada 2 é utilizado. A aresta original a-b é substituída pelas arestas a-c-b.



**Figura 3.4.** Primeira etapa do algoritmo orientado a nível.

Na segunda etapa, é feita uma ordenação dos nós que minimiza os cruzamen-

tos das arestas entre camadas. Na terceira etapa, uma disposição ótima é conseguida arrumando-se os nós sem mudar a ordenação da etapa 2. A disposição equilibrada é encontrada minimizando-se as distâncias entre nós conectados.

Entre as vantagens deste tipo de abordagem pode ser destacada a visualização intuitiva conseguida para grafos com muitas arestas na mesma direção e a extração natural de desenhos para dados que representam alguma hierarquia. Além disso, com grafos bipartite consegue-se uma boa disposição desde que cada camada contenha nós do mesmo tipo.

A maior desvantagem das técnicas orientadas a nível é o fato de que grafos não-dirigidos ou que não representem nenhuma hierarquia são desenhados de maneira muito precária. Para resolver esse problema pode-se transformar grafos não-dirigidos em dirigidos. Essa transformação implica em escolher de forma aleatória a direção das arestas. Mas isso modifica a natureza do grafo e altera a interpretação futura do mesmo.

### 3.1.3 Outras Técnicas

As técnicas de **desenho baseadas em fluxo** formam uma classe popular de algoritmos para grafos planares. Nessa modalidade, os desenhos são ortogonais, isto é as arestas são retangulares e as suas curvaturas são minimizadas. Uma visão geral destes algoritmos é apresentada por BATTISTA [G. D. Battista and Tollis, 1999].

Um outro algoritmo popular para visualização de árvore é o algoritmo de **disposição circular** [Eades, 1984] onde a raiz da árvore é colocada no centro do plano e todos os descendentes são posicionados no círculo concêntrico em torno do nó da raiz. Há ainda um método chamado *Factor-Based Drawing* que pode ser aplicado a grafos em geral e usa critérios de otimização global no desenho de grafos [Niggemann, 2001].

Alguns autores combinaram o desenho de grafos com técnicas de **aglomeração** (clustering). DUNCAN [C. Duncan and Kobourov, 1998], apresenta um método que subdivide o grafo em partes iguais para aglomeração de grafos já desenhados. Uma outra aproximação clássica da otimização em aglomerações foi combinada com a técnica spring embedder [Sablowski and Frick, 1996].

### 3.2 CLASSIFICAÇÃO DAS TÉCNICAS DE VISUALIZAÇÃO

A maioria dos métodos apresentados podem ser agrupados em duas classes [Niggemann, 2001]: Alguns métodos usam informações locais (por exemplo, a área a ser desenhada) para encontrar uma boa disposição (por exemplo, o força-dirigida). Uma vez que negligenciam informações globais, a disposição resultante mostra-se satisfatória em relação aos critérios locais, mas é fraca na ênfase à estrutura global do grafo. Essa classe de métodos é chamada de “Critério de Qualidade” (*Quality Criterion*).

Outros algoritmos usam se enquadram numa classe chamada “Classes de Grafos” (*Graph Class*) que explora o conhecimento sobre a estrutura do grafo. Esses métodos podem ser aplicados somente a classes específicas de grafos como árvores e grafos bipartite - Por definição [Szwarcfiter, 1986], um grafo  $\mathbf{G}(\mathbf{V},\mathbf{E})$  é bipartite quando o seu conjunto de vértices puder ser particionado em dois subconjuntos  $\mathbf{V1}$ ,  $\mathbf{V2}$  pertencentes a  $\mathbf{V}$  tais que toda aresta de  $\mathbf{G}$  une um vértice de  $\mathbf{V1}$  a outro vértice de  $\mathbf{V2}$ .

A Tabela 3.1 a seguir mostra um resumo sobre a classificação dos principais algoritmos abordados. É importante ressaltar que as duas classes citadas não são excluídas.

**Tabela 3.1.** Resumo sobre os principais algoritmos abordados

<b>Algoritmo</b>	<b>Quality Criterion</b>	<b>Graph Class</b>
Força-dirigida	Tamanho das arestas	Grafos em Geral
MDS	Distância entre os nós	Grafos em Geral
Orientado a nível	Hierarquia inerente de um grafo	Grafos Dirigidos
Disposição Circular	Altura da árvore de expansão de nós	Grafos em Geral
Factor-Based	Variâncias	Grafos em Geral
Baseado em fluxo	Direção das Arestas	Grafos Planares

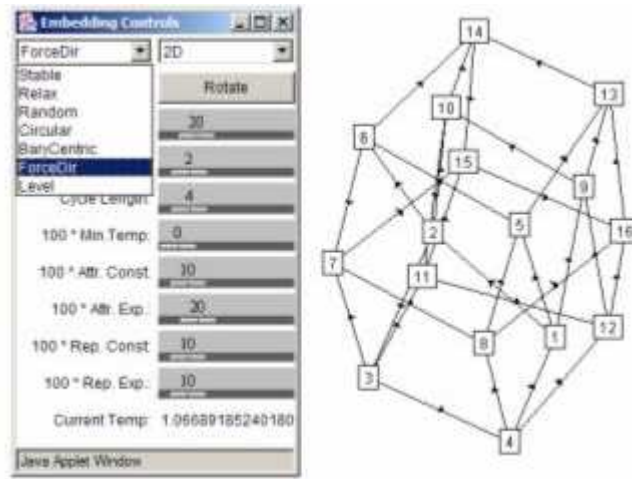
### 3.3 FERRAMENTAS PARA VISUALIZAÇÃO BASEADA EM GRAFOS

Diversos trabalhos têm sido desenvolvidos para a classe de sistemas visualização de informações baseados em grafos. Alguns deles diferem no modelo de arquitetura adotado ou nos algoritmos usados para construção visual. Porém, de alguma forma, todos implementam a visualização das relações entre os dados. Esta seção descreve algumas dessas ferramentas.

#### 3.3.1 IGD (Interactive Graph Drawing)

Um sistema denominado IGD [Erlingsson and Krishnamoorthy, 1996] é um software desenvolvido pelo Departamento de Ciência da Computação do Rensselaer Polytechnic Institute em Nova York - EUA, que permite a construção visual de grafos de forma interativa na Web. O sistema implementa vários algoritmos para desenho de grafos incluindo força-dirigida e orientado a nível.

O IGD permite um alto grau de interação do usuário com a cena visual e contempla operações que vão desde a escolha do algoritmo utilizado até funções básicas de visualização, como rotação e aproximação. A entrada de dados é feita manualmente. A ferramenta foi utilizada para visualização de hierarquias e de makefiles - arquivos que controlam a dependência acíclica de outros arquivos usados na compilação de um software. A Figura 3.5 mostra um grafo visualizado na ferramenta e os seus controles de interação.



**Figura 3.5.** Visualização de grafo usando o IGD.

### 3.3.2 GRAPPA

O GRAPPA é um arcabouço (framework) escrito em Java para visualização de grafos que simplifica o processo de apresentação visual, bem como a manipulação de características específicas dentro de aplicações [Gansner and North, 1999]. Ele implementa os algoritmos da família força-dirigida e orientado a nível. Além disso, ele contempla uma série de funcionalidades e características importantes para visualização de dados como os métodos para construir, manipular, navegar e mostrar grafos com todos os seus elementos (nós, arestas e possíveis subgrafos).

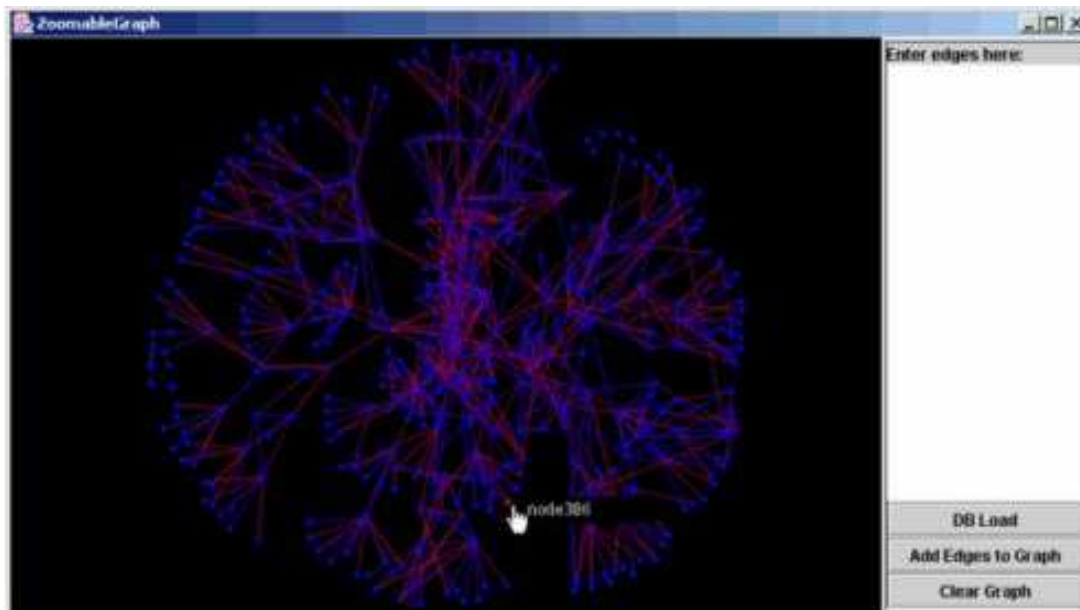


**Figura 3.6.** Visualização de grafo usando o GRAAPA.

Determinados atributos do grafo podem afetar a posição e a aparência de um elemento quando mostrados na tela. Os grafos podem ser criados diretamente pelo uso de métodos disponíveis para criação de elementos. Pode-se ainda, suprimir, selecionar e apagar elementos do grafo de forma interativa. Uma grande quantidade de formas geométricas de nós está disponível. A Figura 3.6 mostra um exemplo da visualização de grafos usando o GRAAPA.

### 3.3.3 Zoomgraph

Enquanto modelos de grafos são aplicados às mais variadas áreas, pesquisadores esforçam-se com as ferramentas que são, muitas vezes, ou demasiadamente gerais, ou muito específicas para suas necessidades. O sistema escrito em Java e denominado Zoomgraph [Adar and Tyler, 2003], permite que usuários possam definir grafos com atributos complexos de nós e arestas e fornece uma linguagem de script para manipular e explorar estes desenhos. A ferramenta, que é pública, inclui também um sistema de visualização baseado em aproximação que permite aos usuários uma movimentação fácil entre as diversas visões do grafo. O software implementa os algoritmos: força-dirigida, disposição circular e orientado a nível. A Figura 3.7 mostra um grafo gerado pelo do Zoomgraph.



**Figura 3.7.** Visualização de grafo usando o ZoomGraph.

### 3.3.4 KARTOO

A ferramenta Kartoo [Kartoo, 2004] é uma interface relacional e visual para um motor de busca na Web. O software não gera como resultado da busca a tradicional lista ordenada de links. Os resultados são exibidos em um grafo onde as arestas correspondem aos termos comuns e os nós podem ser adicionados e removidos para restringir e facilitar uma inspeção visual. As arestas entre os nós são codificadas por cores, permitindo ao usuário o entendimento de como os termos relacionam sites diferentes. Os nós são exibidos pelo tamanho, de acordo com a relevância ao termo da busca. A Figura 3.8 exibe a visualização da busca do site UNIFACS através do Kartoo.



**Figura 3.8.** Visualização de grafo resultante da busca através do Kartoo.

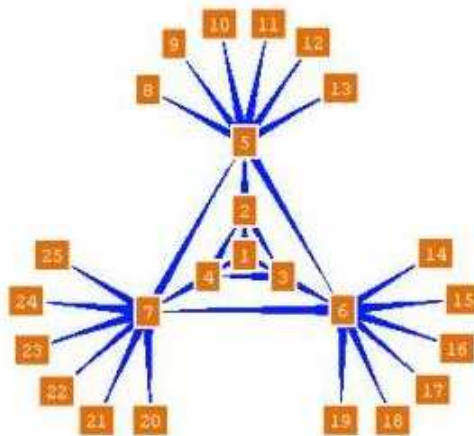
### 3.3.5 TouchGraph

O TouchGraph [TouchGraph, 2004] é um arcabouço escrito na linguagem Java que permite a construção de aplicações para visualizar redes de informação inter-relacionadas tais como em bases de dados relacionais ou documentos hipermídia na Web. As redes são construídas como grafos interativos, que se prestam a uma variedade de atividades. Acoplado a uma imagem, um usuário pode navegar através de grandes redes e explorar maneiras diferentes de arrumar os seus componentes na tela, realizando operações como aproximação, rotação e navegação nos diferentes níveis do grafo.



Navegar em um grafo gerado por um software que utiliza este arcabouço é inerentemente um processo dinâmico cujas etapas necessitam ser feitas passo a passo para manter a atenção do usuário orientada e sob controle. O TouchGraph consegue isso mantendo a aparência do grafo tão estática quanto possível. Mais importante, ele se certifica de que as mudanças dinâmicas são intuitivas, repetitivas e passíveis de serem desfeitas. A natureza associativa de um grafo permite lembrar sua estrutura de forma fácil, mas é a experiência de ver uma série de imagens visuais que dão impulso à capacidade do ser humano em absorver melhor a informação mostrada.

O arcabouço tem código fonte aberto e implementa os algoritmos de força-dirigida. Uma série de aplicações usa este arcabouço como base da sua construção. O TouchGraph AmazonBrowser [Amazon, 2004], permite a inspeção visual das relações de compras similares no site da livraria Amazon. Os itens disponíveis para compra são ligados por arestas dirigidas, onde uma aresta A-B indica que uma pessoa que compra A também compra B. O TouchGraph GoogleBrowser [Google, 2004], o TG WikiBrowser [WikiBrowser, 2004] e o Planet Navigator [Planet, 2004] são motores visuais de busca por informações na Web que usam o TouchGraph. Redes sociais de relacionamentos na Web podem ser visualizadas usando o LIVEJOURNAL [Livejournal, 2004]. A Figura 3.9 mostra a visualização de um grafo usando o TouchGraph.



**Figura 3.9.** Visualização de grafo através do TouchGraph.

### 3.3.6 Outras Ferramentas

Arcabouços de análise tais como o Ucinet [S.Borgatti, 2004a] são usados para dividir o grafo e na execução de vários cálculos sobre a sua estrutura. A saída destes arcabouços é passada então a outros como Pajek [Batagelj and Mrvar, 1997], Netdraw [S.Borgatti, 2004a], GraphViz [Gansner and North, 1999], que são poderosos para desenhar e executar operações de disposição e, por necessidade, tendem a atender somente aos requisitos mais comuns para visualização de grafos (cores, formas, etc.). Explorar um grafo baseado em uma ou mais das complexas propriedades de nó é uma etapa do processo de inspeção visual que está entre a análise e a visualização.

Alguns autores tratam este problema gerando uma API (application program interface) comum para a manipulação de objetos de um grafo tanto em memória quanto visualmente [M. S. Marshall and Malancon, 2000] [yWorks, 2004]. Porém, esta abordagem requer compilação e não permite que o usuário interaja com o grafo dinamicamente. Para dar suporte a tal comportamento faz-se necessária uma infra-estrutura adicional que forneça o acesso (ou o armazenamento) a atributos complexos de objetos do grafo. Um sistema denominado Libsea [Libsea, 2003] é um conjunto de classes Java que fornece funções de armazenamento projetado para manipular objetos do grafo com uma grande quantidade de atributos. Entretanto, ainda não fornece funções de visualização ou uma linguagem de manipulação específica para o grafo.

## 3.4 O CENÁRIO ATUAL DAS FERRAMENTAS DE MINERAÇÃO VISUAL BASEADAS EM GRAFOS

Conforme visto na seção anterior, vários protótipos, modelos de arquitetura e ferramentas foram desenvolvidos a partir das pesquisas relacionadas ao tema ferramentas de visualização usando grafos.

Com base na revisão de literatura realizada neste trabalho, um resumo do cenário atual das ferramentas de mineração visual baseadas em grafos é apresentado a

seguir. As ferramentas são agrupadas baseadas nos seguintes critérios: área de aplicação ou objetivo; família de algoritmos utilizados (ver na seção 3.2 a Tabela 1); controles de visualização; e finalmente o tipo de entrada dados suportada.

### 3.4.1 Áreas de Aplicação

Algumas áreas de aplicação mostram-se particularmente interessantes para utilização das ferramentas citadas anteriormente. Na área de desenho de grafos, as ferramentas ou modelos são utilizados apenas para o exibir um grafo em duas ou três dimensões na tela de um computador [Erlingsson and Krishnamoorthy, 1996] [Gdt, 2004] [Ilog, 2004] [Gansner and North, 1999] [Graphlet, 2004] [aiSee, 2004]. Nesse caso, a maioria das ferramentas exibe disposições de grafos cujo objetivo principal é o desenho do grafo.

As ferramentas na área de visualização de informação, além de exibir o desenho do grafo, mostram disposições que destacam as relações existentes entre as informações. Isto permite que um perito no domínio dos dados consiga obter informações desconhecidas ou implícitas até então.

Na área de visualização de informações, pode-se definir as algumas subáreas nas quais são agrupadas uma série de ferramentas:

- i) Web [Amazon, 2004] [Google, 2004] [Kartoo, 2004] [Livejournal, 2004] [Planet, 2004]
- ii) Bancos de dados relacionais [Adar and Tyler, 2003] [ThinkMap, 2004].
- iii) Softwares [Gansner and North, 1999] [aiSee, 2004] [Ilog, 2004].
- iv) Redes de computadores [D. Krackhardt and Mcgrath, 1995] [Batagelj and Mrvar, 1997] e Redes sociais [Livejournal, 2004].

### 3.4.2 Algoritmos Utilizados

As ferramentas que têm como objetivo principal obter desenhos que sejam intuitivos a um usuário utilizam a família de algoritmos força-dirigida. Essa família de algoritmos mostra-se muito eficiente no desenho de disposições que se assemelham com a noção estética de um ser humano para grafos, apesar de apresentar uma complexidade elevada e de não ser ideal para visualização de grafos com muitos nós, como visto na seção 3.1.1. Esse algoritmo é adotado na maior parte das ferramentas citadas [D. Krackhardt and Mcgrath, 1995] [Batagelj and Mrvar, 1997] [Gansner and North, 1999] [Adar and Tyler, 2003] [ThinkMap, 2004] [aiSee, 2004] [S.Borgatti, 2004a] [Planet, 2004] [Ilog, 2004] [TouchGraph, 2004].

Outras ferramentas cujo objetivo é visualizar grafos com muitos nós ou grafos que representam alguma hierarquia fazem uso dos algoritmos Orientados a nível e Circular em complemento aos algoritmos de força-dirigida [Erlingsson and Krishnamoorthy, 1996] [Gansner and North, 1999] [Adar and Tyler, 2003].

### 3.4.3 Controles de visualização

A existência de controles de visualização em uma ferramenta de mineração visual baseada em grafos indica a possibilidade de interação do usuário com a cena visual. A maior parte das ferramentas estudadas apresenta controles que permitem realizar operações como aproximação, rotação e re-posicionamento de elementos que compõem o grafo. Além disso, é possível modificar atributos visuais destes elementos como cor, formato, tamanho e etc [Erlingsson and Krishnamoorthy, 1996] [Gansner and North, 1999] [Adar and Tyler, 2003] [ThinkMap, 2004] [Livejournal, 2004] [Amazon, 2004] [Google, 2004] [WikiBrowser, 2004] [Planet, 2004] [TouchGraph, 2004].

### 3.4.4 Entrada de dados

Os tipos de entrada de dados que são suportados pelas ferramentas estudadas podem ser agrupados em:

- i) Entrada através de um banco de dados relacional onde os dados são obtidos diretamente do banco [Adar and Tyler, 2003] [ThinkMap, 2004].
- ii) Entrada de dados manual onde as informações que compõem a estrutura de um grafo como os nós e as arestas são fornecidos diretamente pelo usuário [TouchGraph, 2004] [Erlingsson and Krishnamoorthy, 1996] [Gansner and North, 1999].
- iii) E finalmente, entrada de dados através da Web onde as informações são provenientes de documentos hipermídia espalhados pela Internet [Amazon, 2004] [Google, 2004] [Kartoo, 2004] [WikiBrowser, 2004] [Planet, 2004] [Livejournal, 2004].

## 3.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado um resumo de pesquisas e trabalhos encontrados durante a revisão da literatura do tema visualização de informação baseadas em grafos. Um cenário atual sobre as ferramentas e modelos relacionados ao tema foi resumida com base no estudo realizado neste trabalho. O próximo capítulo descreve como os dados são armazenados em bancos de dados relacionais e como é feito o mapeamento desses dados para um grafo.

# O DESENHO DE GRAFOS A PARTIR DE BANCOS DE DADOS RELACIONAIS

A utilização das técnicas de visualização de dados baseada em grafos vista no capítulo 3, exige que os dados estejam estruturados. Como nem sempre os dados estão neste formato, as ferramentas para mineração visual de dados relacionais fazem antes o pré-processamento dos dados. Pré-processar dados de forma a transformá-los em dados estruturados não é uma tarefa simples. Segundo MENDONÇA [Mendonça, 2001], é um grave erro considerar o pré-processamento dos dados como uma tarefa secundária.

Uma abordagem interessante para visualizar dados em grafos é obter os dados no formato estruturado sem a necessidade de pré-processamento. Hoje em dia, os mecanismos mais comuns para o armazenamento de dados estruturados são bancos de dados relacionais. Um sistema de gerenciamento de banco de dados relacionais (SGBD) pode ser definido como um conjunto de dados associados a uma série de programas para acessar esses dados [A. Silberschatz and Sudarshan, 1999].

Os dados armazenados em um SGBD são organizados em estruturas que permitam a sua recuperação de forma rápida e eficiente. Estas estruturas estão divididas em 3 níveis [Date, 1986]: físico, lógico e de visão. O nível físico é o mais baixo nível de abstração que descreve como os dados estão de fato armazenados. O nível lógico descreve quais são os dados armazenados e quais são as relações entre eles. O nível de visão é o mais alto nível de abstração e descreve apenas uma visão parcial daquilo que representa o banco de dados.

No nível lógico de um banco de dados estão os repositórios onde são encon-

tradas informações que permitem a representação visual dos dados e suas relações. A obtenção da estrutura dos dados dentro de um banco relacional é possível através dos metadados que descrevem os dados. Este capítulo apresenta alguns conceitos básicos sobre bancos de dados, metadados, modelo relacional e, finalmente, a nossa abordagem de mapeamento de dados do modelo relacional para grafos.

## 4.1 BANCOS DE DADOS E METADADOS

Um banco de dados pode ser entendido como uma coleção organizada de informações que estabelece relações naturais entre dados. Segundo CHIAVENATTO [Chiavenatto, 1992], um banco de dados é um sistema de armazenamento de dados devidamente codificados e disponíveis para o processamento e obtenção de informações. É um conjunto de arquivos, relacionados logicamente, organizados de forma a melhorar e facilitar o acesso aos dados e ainda, eliminar a redundância.

Para DATE [Date, 1986], um sistema de gerenciamento de banco de dados (SGBD) é uma forma de gerenciar o armazenamento de dados em sistemas computacionais, ou seja, é um mecanismo cujo objetivo global é registrar e manter informações.

Um banco de dados armazena não só os dados, mas também as informações sobre esses dados. Estas informações sobre os dados recebem o nome formal de metadados e estão no nível lógico do banco. Para DATE [Date, 1986], metadados são informações que resumem, enriquecem ou complementam os dados referenciados, produzindo assim um potencial incremento de informação. A Figura 4.1 mostra o modelo proposto por CHIAVENATTO [Chiavenatto, 1992] para relacionar a informação com seus metadados:

- i) **Embutidos:** Os metadados estão descritos nos próprios dados. Um exemplo disso são documentos HTML. Em HTML é possível, através de uma “tag”, a especificação de qualquer elemento de metadados, o que poderá ser ignorado pelas máquinas no processamento de HTML, mas pode ser aproveitado por outros sistemas que identifiquem esses elementos.

- ii) **Associados:** Os metadados estão descritos nos mesmos repositórios que os dados mas em áreas diferentes. Dados em bases de dados relacionais por exemplo, guardam os metadados relativos às relações entre seus registros em tabelas diferentes daquelas que armazenam dados.
- iii) **Separados:** Os metadados estão descritos em repositórios separados dos dados. Um exemplo clássico é o modelo de biblioteca tradicional, onde há bases de dados bibliográficas existentes em sistemas próprios e os dados, isto é, os livros, ficam nas prateleiras.

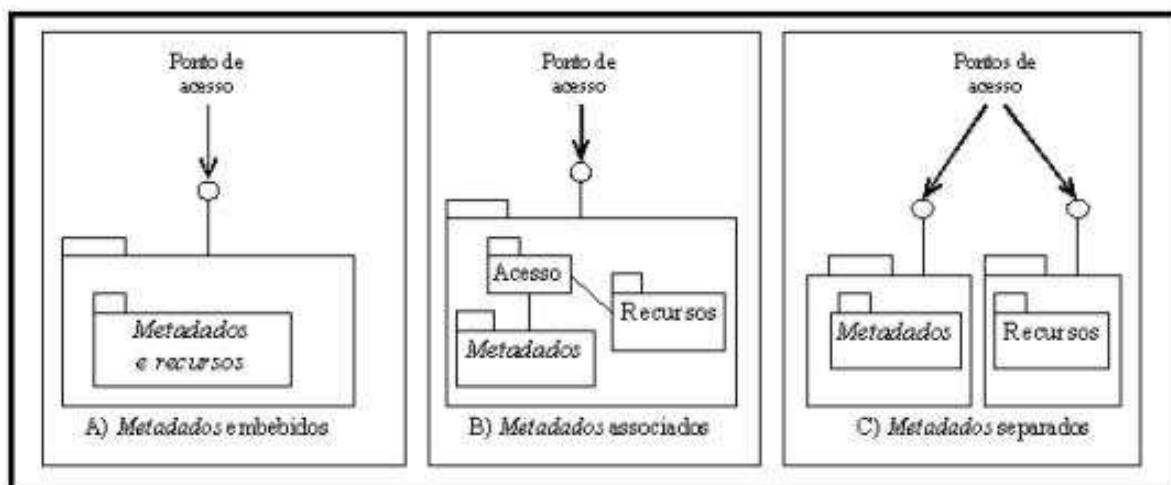


Figura 4.1. Relacionamento entre dados e metadados.

## 4.2 O MODELO RELACIONAL

O modelo relacional é definido como uma percepção que mapeia os objetos do mundo real como entidades (tabelas) e os relacionamentos entre esses objetos como relações. Então, um banco de dados relacional consiste em uma coleção de tabelas e relações baseadas no modelo relacional, cada uma das quais com um nome único [A. Silberschatz and Sudarshan, 1999].

Uma tabela pode ser vista como uma estrutura onde cada linha (tupla) pode representar um conjunto de registros e cada campo (coluna) define uma das suas características [Date, 1986]. A Figura 4.2 exibe um exemplo de uma tabela de nome "Cafés".



Ela possui 5 colunas: “Tipo de Café”, “ID-Vendedor”, “Preço”, “Vendas” e “Total”. Segundo a terminologia descrita pelo modelo relacional de dados [Chiavenatto, 1992], essas colunas recebem o nome de atributos. Cada um desses atributos contém uma lista de valores de um mesmo domínio. Por exemplo, a coluna “Preço” contém os valores relativos ao preço do café.

campo  
↓

Tabela “Cafés”

registro →	Tipo de Café	ID_Vendedor	Preço	Vendas	Total
	Colombiano	1001	1,35	0	0
	Francês	201	1,38	0	0
	Expresso	201	1,43	0	0
	Brasileiro	1001	1,31	0	0
	Descafeinado	201	1,33	0	0

**Figura 4.2.** Exemplo de dados organizados em uma tabela.

Uma tabela se relaciona com uma ou mais tabelas através dos valores de suas colunas. Quando um relacionamento é definido, uma coluna de uma tabela é associada com a coluna de outra tabela. As informações sobre esses relacionamentos são armazenadas também em tabelas. Essas tabelas especiais armazenam o nome das tabelas relacionadas e dos seus respectivos campos que são metadados do tipo associados, conforme visto na seção 4.1. A Figura 4.3 (a) exhibe a relação entre as tabelas “Filmes”, “Produtora” e “Gênero” de um banco relacional. A Figura 4.3 (b) mostra a tabela que armazena informações sobre os relacionamentos das tabelas desse banco

O modelo prevê também, a existência de linguagens relacionais para manipulação dos dados e metadados de um banco. Essas linguagens suportam operações como seleção, inserção e alteração dos registros de uma ou mais tabelas de um banco relacional. Existem diversas linguagens para manipulação de bancos de dados entre as quais: QBE (*Query-by-Example*), Quel, Datalog e SQL. A SQL (*Structured Query Language*) foi criada originalmente pela IBM em 1970 e é hoje a mais utilizada dessas linguagens [SQL-92, 2004].



(a)

szColumn	szObject	szReferencedColumn	szReferencedObject	szRelationship
PRODUTORA	FILMES	PRODUTORA_NOME	PRODUTORA	PRODUTORAFILMES
GENERO	FILMES	GENERO_NOME	GENERO	GENEROFILMES

Registro: 1 de 2

(b)

**Figura 4.3.** (a) Relacionamentos entre as tabelas de um banco relacional. (b) Tabela que armazena as informações sobre os relacionamentos.

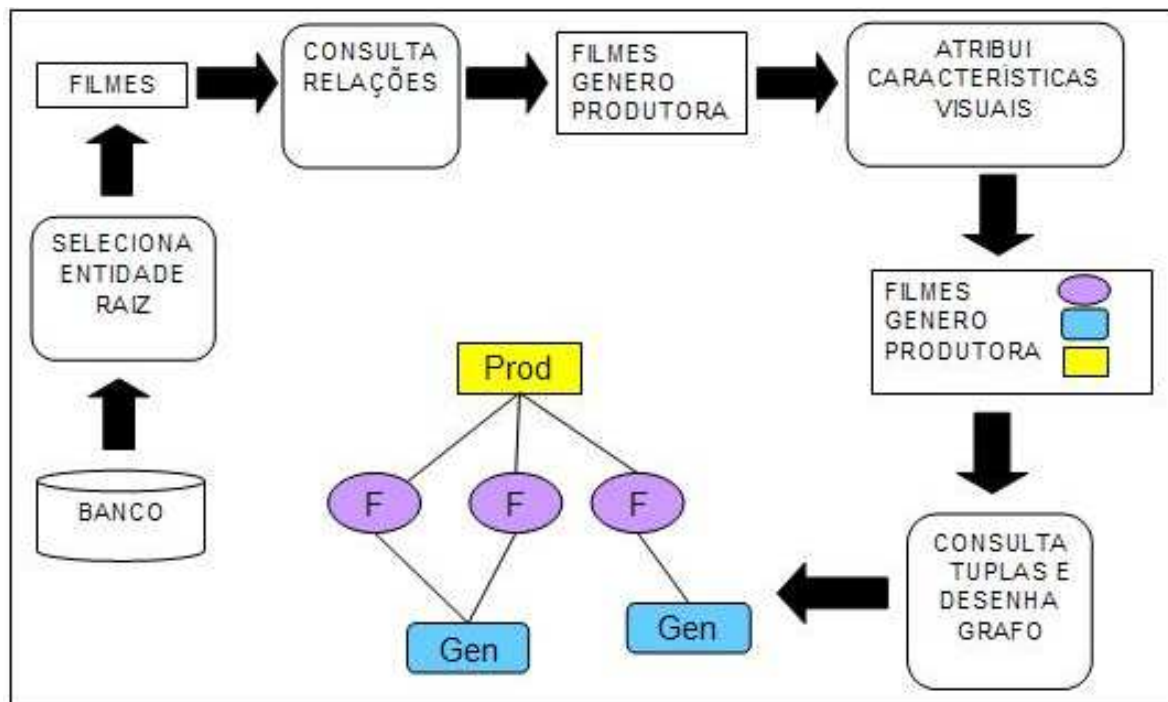
### 4.3 O MAPEAMENTO DE DADOS EM BANCOS RELACIONAIS PARA GRAFOS

Uma boa representação visual dos dados armazenados em bancos relacionais passa pela escolha da metáfora visual adequada. Um grafo é uma boa metáfora para dados que têm relações inerentes pois, pode-se mapear os dados como nós e as relações entre eles como arestas. Logo, grafo torna-se uma metáfora visual interessante para visualização de dados armazenados em bancos de dados relacionais. Para se construir um grafo a partir de um banco relacional é necessário seguir alguns passos:

- i) Deve-se escolher uma entidade (entidade raiz) do banco que se quer visualizar;
- ii) O próximo passo é descobrir quais são as entidades que se relacionam com a entidade raiz. Isto é feito, através de uma consulta que recebe como parâmetro o nome da entidade raiz e é feita na tabela que guarda as informações sobre as relações do banco (ver Figura 4.3 (b));
- iii) Depois disso, deve-se selecionar as tuplas da entidade raiz e de cada uma das entidades relacionadas;

- iv) Para cada tupla da entidade raiz, existe uma ou mais tuplas relacionadas. Todas estas tuplas devem ser mapeadas como nós do grafo e as relações entre elas como arestas.

A Figura 4.4 exibe a construção de um grafo a partir do banco de dados relacional mostrado na Figura 4.3 (a). Primeiro a entidade “filme” é selecionada como raiz. Depois disso, descobre-se as entidades relacionadas: gênero e produtora. Para facilitar a visualização, deve-se selecionar atributos visuais que serão associados com cada tipo de entidade. Na Figura 4.4, produtoras são mostradas como retângulos, os filmes como elipses e os gêneros de filmes como retângulos de bordas arredondadas. As tuplas das entidades de filme, gênero e produtora são selecionadas e mapeadas como nós. Para cada tupla da entidade “filme”, seleciona-se uma tupla relacionada de “gênero” e outra “produtora”. Para cada par de tuplas “filme-gênero” e “filme-produtora” cria-se uma aresta.



**Figura 4.4.** Passos para construção de um grafo a partir de um banco de dados relacional.

## 4.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado um resumo da nossa abordagem para visualização de informações armazenadas em bancos de dados relacionais através de grafos. Uma visão geral sobre os mecanismos de armazenamento baseado no modelo relacional de dados foi apresentada junto com os passos para o desenho de grafos a partir de dados armazenados neste formato. O próximo capítulo descreve detalhadamente a ferramenta GraphMiner, seus requisitos sua arquitetura e sua implementação.

# A FERRAMENTA GRAPHMINER

Segundo ADAR [Adar and Tyler, 2003] a popularidade de modelos visualizados como grafos tem aumentado em domínios variados como as ciências sociais, comportamento organizacional, física, e ciências biológicas. Enquanto isso, o número de opções de ferramentas de visualização disponíveis aos pesquisadores ainda é muito reduzido. Estas ferramentas são, muitas vezes, ou demasiadamente gerais para manipular modelos específicos de grafos, ou limitadas a um determinado domínio.

A ferramenta GraphMiner foi criada para permitir que um usuário defina, descreva, e visualize dados e suas relações utilizando a metáfora visual de um grafo. A ferramenta é particularmente intuitiva aos usuários com pouca experiência em informática, contudo fornece opções avançadas de consultas visuais e acesso a dados diretamente de bases de dados relacionais.

Como descrito no final do capítulo 4, a GraphMiner mapeia registros de tabelas para nós e as relações entre as tabelas para arestas. Assim, os nós e arestas representados podem ter diferentes atributos associados a eles. Estes atributos correspondem aos campos dos registros mapeados nos nós e arestas do grafo.

Os usuários da GraphMiner podem especificar, através de uma vasta opção de controles de interação, as associações de atributos visuais (cor, formato, rótulo e etc.) com atributos reais do banco de dados relacional. Isso permite que um perito no domínio de uma determinada aplicação explore os dados de forma natural e intuitiva.

Os usuários da ferramenta GraphMiner são, a princípio, os peritos em domínios

de específicos de uma organização. Um dos objetivos da ferramenta é obter relações de interesse entre dados que nem sempre estão visíveis. As informações sobre as relações entre dados podem ser úteis a uma grande gama de profissionais, especialmente para aqueles cujo trabalho depende do entendimento de como as informações armazenadas no banco de dados estão organizadas em rede.

Uma vez que as informações estão armazenadas em uma base de dados acessível pela ferramenta, é necessário permitir que estas informações sejam consultadas da maneira mais adequada a cada situação. Para isso, a representação visual das relações entre os dados é disponibilizada, bem como a navegação através dos seus elementos, definidos a partir de diversas formas de pesquisa sobre o conteúdo da base de dados. Estas pesquisas são implementadas por mecanismos que permitem a modificação de atributos visuais do modelo visualizado baseada nos valores reais contidos na base de dados.

## **5.1 REQUISITOS FUNCIONAIS DA FERRAMENTA**

Para que os objetivos definidos para a ferramenta GraphMiner sejam alcançados se faz necessária a definição de alguns dos requisitos funcionais relacionados à visualização de informações baseada em grafos, conforme descrito a seguir:

- i) Apoiar o processo de visualização de informações relacionais:
  - a. Mostrar relacionamentos através de grafos.
  - b. Permitir a atribuição dinâmica de atributos reais a atributos visuais.
  - c. Permitir navegação sobre os grafos.
  - d. Permitir transformações interativas sobre a cena visual (ex: aproximação, rotação, translação).
- ii) Apoiar o processo de descoberta de conhecimento de dados relacionais:
  - a. Visualizar relações (vide item 1).

- b. Realizar consultas interativas através de widgets (caixas de texto, rótulos, janelas e outros componentes de interface gráfica para visualizar e filtrar dados).
  - c. Realizar buscas por palavras chave.
  - d. Exibir detalhes sobre demanda.
- iii) Permitir a conexão e extração transparente de relações e dados relacionais de sistemas de gerenciamento de banco de dados que usam o modelo relacional
- iv) Navegar tanto sobre os dados como sobre o esquema da base de dados.

## 5.2 DESENVOLVIMENTO DA FERRAMENTA

A ferramenta implementa modelo de software de três camadas (apresentação, negócio e acesso a dados), as quais estão logicamente separadas e representam uma abstração do domínio de um problema específico. A camada de apresentação é definida pelas classes que compõem a interface gráfica para utilização do usuário. A camada de negócio é formada por classes que implementam as regras para a construção visual de um grafo que represente as relações entre as informações contidas em um banco de dados. A camada de acesso a dados é a única a ter conhecimento de como e onde os dados estão armazenados, ficando responsável por recuperar estes dados em diversos formatos.

A ferramenta GraphMiner foi desenvolvida na plataforma Java [Sun Microsystems, 2003] que exibe as vantagens proporcionadas pela orientação a objetos, pela portabilidade do código produzido e pela facilidade de integração com diversos ambientes.

## 5.3 ARQUITETURA DA FERRAMENTA

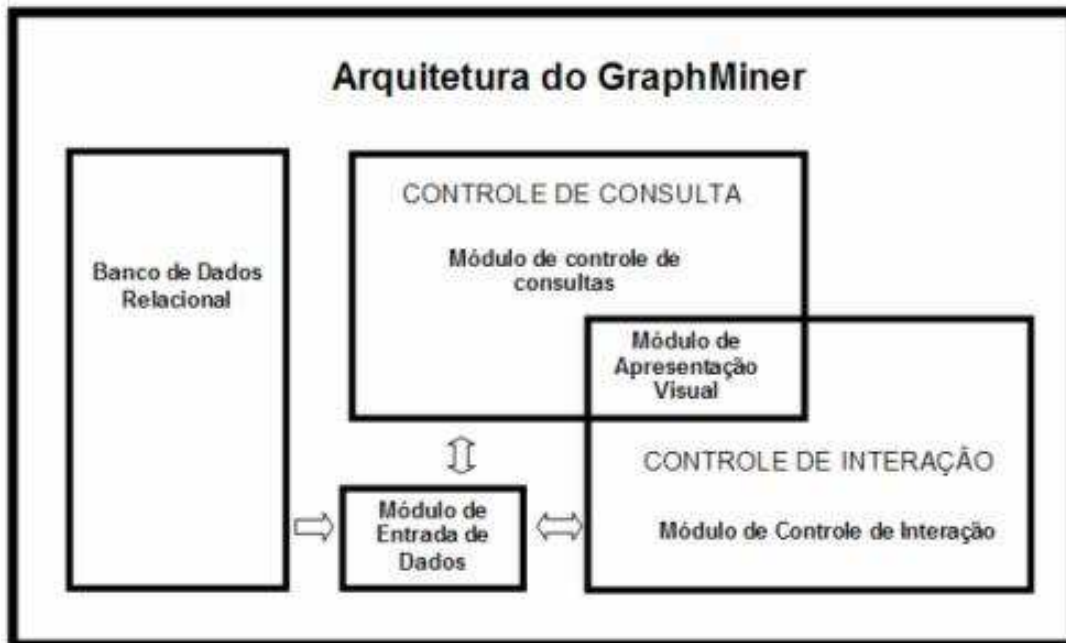
A ferramenta da GraphMiner é dividida em quatro módulos inter-relacionados:

- i) Módulo de entrada de dados.
- ii) Módulo de apresentação visual.

- iii) Módulo de controle de consulta.
- iv) Módulo de controle de interação.

Estes módulos são responsáveis pela obtenção das informações armazenadas em um banco de dados, pela criação de uma interface que possibilita realizar consultas sobre os dados, pela seleção, filtragem e modificação de atributos dos elementos que compõem o grafo e pela representação visual em duas dimensões do grafo na tela do computador. A arquitetura da ferramenta é apresentada na Figura 5.1.

O módulo de entrada de dados se relaciona a uma base de dados relacionais. Conforme visto na seção 4.1 estas bases possuem estruturas predefinidas para armazenamento de seus metadados. O módulo de entrada de dados é responsável pelo acesso às informações (Dados e metadados) da base que são passadas aos módulos de controle de consultas e de interação. A partir das informações passadas, os módulos de consulta e interação fornecem subsídios ao módulo de representação visual para construção do grafo.



**Figura 5.1.** Arquitetura da ferramenta GraphMiner.

A ferramenta GraphMiner tem seus módulos implementados a partir do agrupamento de classes em pacotes de software. Estes componentes são unidades autocontidas, independentes e reutilizáveis que interagem através de suas interfaces públicas. Desta



forma, a lógica de funcionamento da ferramenta está dividida entre os componentes dos seus quatro módulos, visando simplificar o seu projeto e implementação.

A seguir a descrição detalhada de cada módulo, com destaque às suas principais funcionalidades.

### 5.3.1 Módulo de Entrada de Dados

O módulo de entrada de dados foi desenvolvido com o objetivo de permitir a integração da GraphMiner aos diversos tipos de bancos de dados relacionais existentes. Esta integração proporciona um aumento na quantidade das fontes de informação que podem ser utilizadas pela ferramenta. Desta forma, a ferramenta pretende cumprir os requisitos funcionais definidos na seção 5.1 e ajudar a descoberta de informações potencialmente úteis e inéditas, até então.

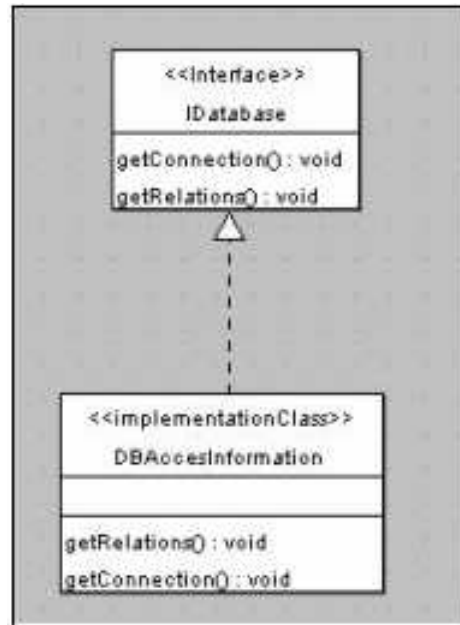
Uma das funcionalidades desse módulo é o processo de carga automática de informações provenientes de bases de dados. Usando esta funcionalidade foram criados alguns componentes para acesso a banco de dados relacionais.

A interface *IDatabase* define uma maneira padrão de estabelecer conexão com uma base de dados. As classes concretas que cumprem essa interface contêm os requisitos necessários para implementar a lógica de acesso a dados em bases relacionais. Para isto, utilizam a API (interface de programação de aplicação) JDBC (*Java Database Connectivity*) [Sun Microsystems, 2003] e uma API específica disponibilizada pelo banco.

Estas APIs oferecem um mecanismo de comunicação a partir do qual é possível realizar consultas em SQL (*Structured Query Language*) cujo resultado fornece informações de entrada do sistema. Essas informações são registros armazenados em tabelas do banco e os metadados que descrevem esses registros. Na ferramenta esses registros são representados como nós do grafo e os metadados, quando definem as relações entre os dados,

são representados por arestas.

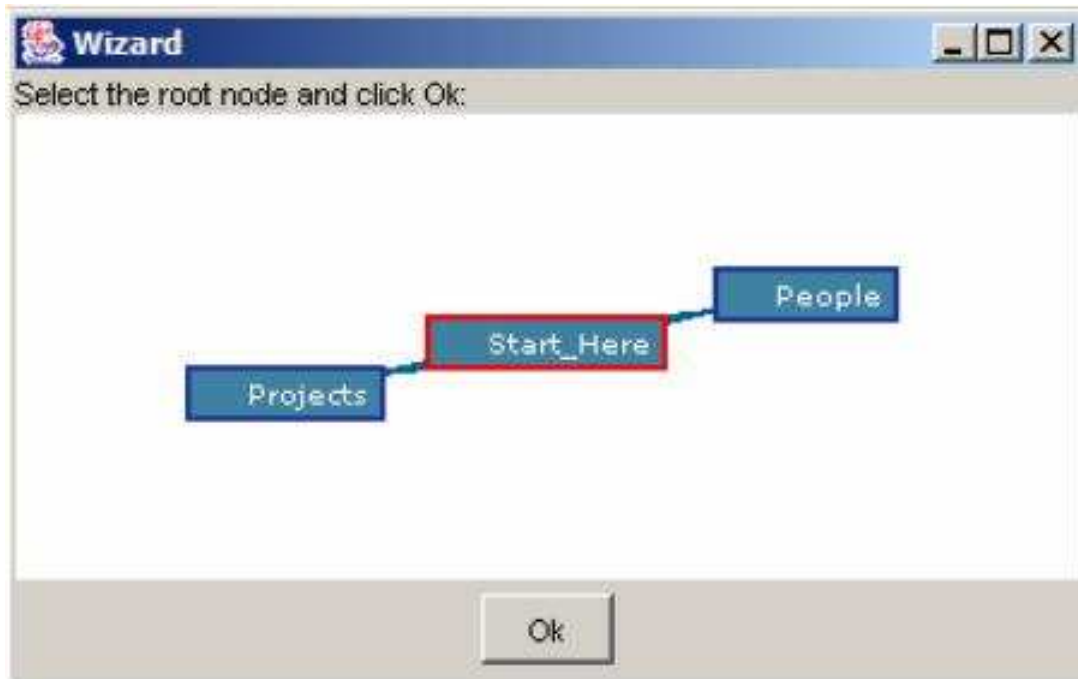
A classe concreta *DBAccessInformation* por exemplo, define os atributos e cumpre os métodos especificados na interface *IDatabase* para acessar bancos de dados Microsoft Access. A Figura 5.2 exibe o diagrama das classes de acesso a dados da ferramenta GraphMiner.



**Figura 5.2.** Diagrama das classes de acesso a dados.

O módulo de acesso a dados usa, além dos componentes citados, outros componentes que permitem a configuração, de forma visual, das informações necessárias à obtenção dos dados. As classes *WizardPanel* e *WizardDB* consultam os metadados e permitem visualização do diagrama de entidade-relacionamento (DER) de um banco de dados relacional. A partir desse diagrama, um usuário pode escolher que entidades devem ser visualizadas pela ferramenta. A Figura 5.3 mostra a visualização do DER de um banco relacional na ferramenta GraphMiner. Neste banco existem três tabelas: *Projects*, *People* e *Start-Here*. A tabela *Start-Here* é uma entidade do tipo relação que representa as relações entre as tabelas *Projects* e *People*. As relações entre as entidades mencionadas são representadas visualmente pelas arestas do grafo

Através do diagrama da estrutura do banco de dados relacional exibido pela ferramenta, o usuário pode selecionar qual será a entidade do banco cujos nós serão



**Figura 5.3.** Visualização de um DER pela GraphMiner.

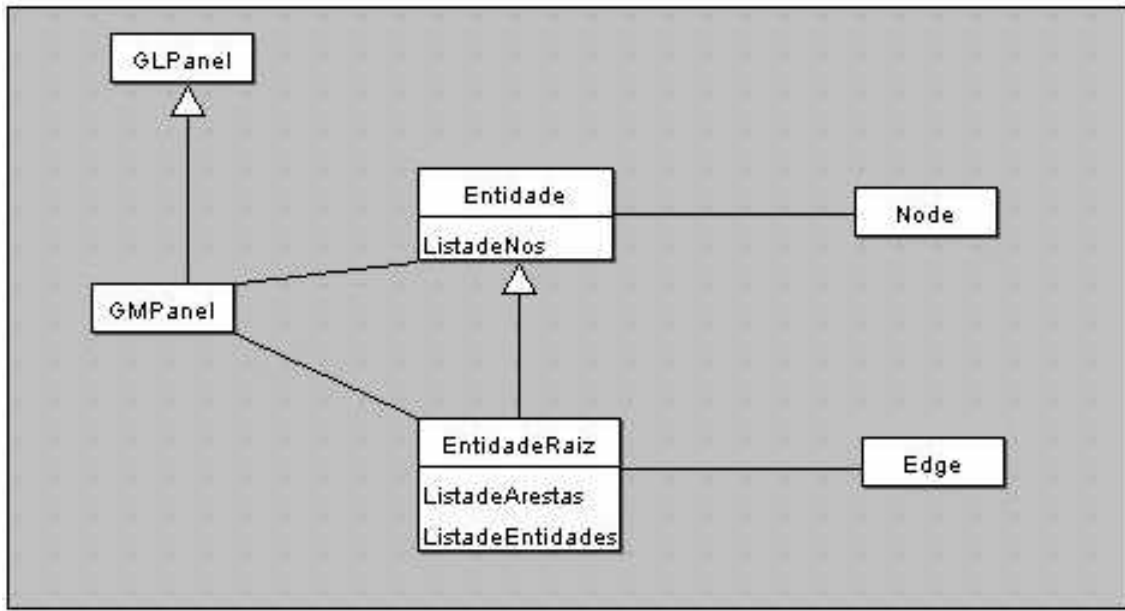
representados como nós raiz. A partir dessa informação o módulo de entrada de dados seleciona todos os registros da entidade selecionada e ainda, os registros relacionados com essa entidade. Todos os registros são passados ao módulo de apresentação visual que realiza a construção do grafo na tela do computador.

### 5.3.2 Módulo de Apresentação Visual

Como dito anteriormente, a técnica de visualização baseada em grafos mostra-se adequada ao processo de visualização de dados em bancos relacionais devido à sua natureza combinatória e geométrica para a representação de informações. No módulo de apresentação visual, cada registro de uma tabela é representado como um vértice do grafo e seus relacionamentos como arestas. Os atributos visuais cor, tamanho, forma e etiquetagem são utilizados para a criação da cena visual. Desta forma, os registros de um banco de dados relacional são representados visualmente, permitindo busca, navegação e mergulho nas informações.

O módulo de apresentação visual utiliza o arcabouço TouchGraph para im-

plementar a representação visual de grafos. O arcabouço é composto por um conjunto de classes e interfaces que fazem a representação de uma coleção de objetos na estrutura visual de um grafo. As classes que compõem a estrutura da ferramenta herdam da classe `GLPanel`, presente neste arcabouço.



**Figura 5.4.** Diagrama das Classes do TouchGraph adaptadas para a GraphMiner.

No desenvolvimento deste trabalho foi necessária uma adequação do TouchGraph para que o grafo pudesse exibir informações específicas dos dados explorados (por exemplo, detalhes sobre demanda). A Figura 5.4 exibe o diagrama de classes das modificações feitas no arcabouço. Os dados de cada entidade são selecionados e compostos em um objeto da classe *EntidadeRaiz*, a qual herda a classe *Entidade*. Desta forma, todos os dados selecionados e seus respectivos relacionamentos são representados como vértices e arestas da estrutura visual.

Cada objeto do tipo *EntidadeRaiz* contém:

- i) Uma lista de objetos da classe *Node* que representa através de nós, os valores de

cada uma das tuplas da entidade.

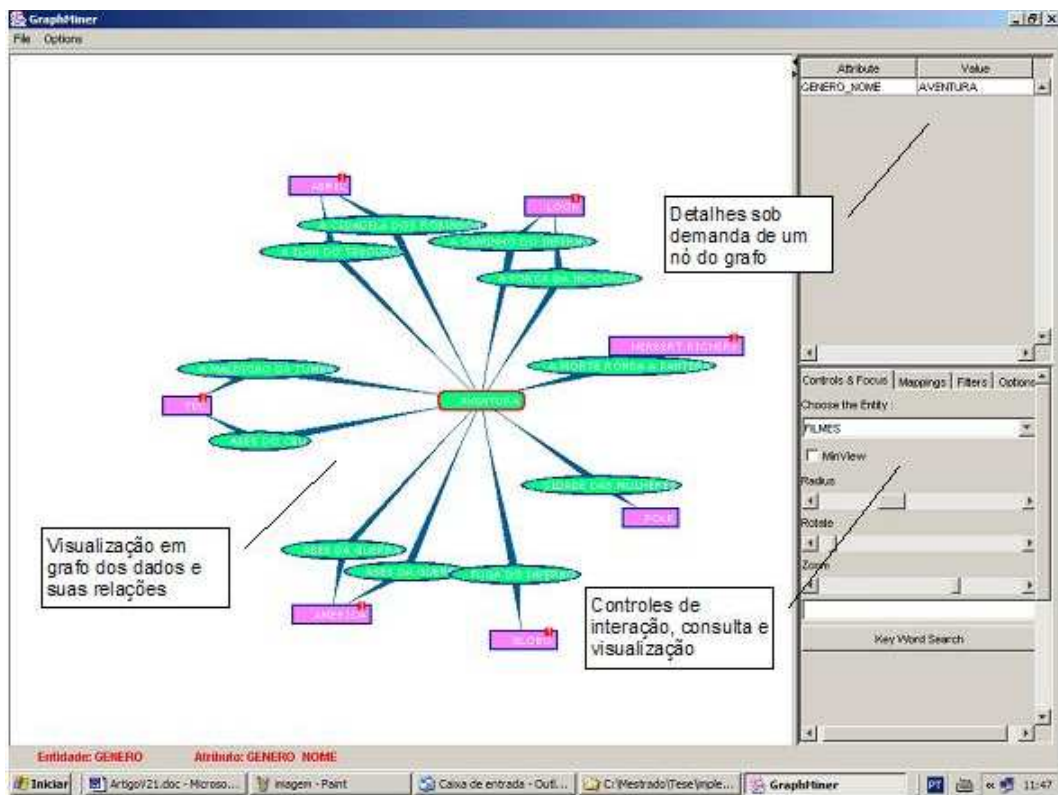
- ii) Uma lista de objetos do tipo *Entidade* que representam suas entidades relacionadas. Cada objeto dessa lista contém também, uma lista de objetos da classe *Node*.
- iii) Uma lista com objetos da classe *Edge* que representa através de arestas, os valores de cada uma das relações entre tuplas das entidades relacionadas com a entidade raiz.

Uma vez que a estrutura de nós e arestas das entidades selecionadas está montada, pode-se realizar a construção visual do grafo. A classe *GMPanel*, herdeira de *GLPanel*, contém um objeto denominado painel que permite o desenho de figuras geométricas como círculos, retângulos e elipses. É nesse painel que o grafo é desenhado. Esse desenho é feito com a simples inserção dos objetos do tipo *Node* e *Edge* contidos na estrutura de entidades mencionada.

Para cada conjunto de nós que representam as tuplas de uma entidade do banco de dados, existe um conjunto de atributos visuais associados como forma, cor e rótulo exibido. O nó selecionado é colocado no centro do painel de visualização e tem a borda colorida de vermelho para diferenciá-lo dos demais. A Figura 5.5 mostra a visualização de dados e suas relações a partir da ferramenta GraphMiner.

Um dos diferenciais da ferramenta desenvolvida no presente trabalho reside no fato desta permitir a inspeção dos dados e de suas relações de maneira intuitiva e visual. Para um usuário com pouca experiência em bancos de dados, obter e navegar sobre os dados se torna uma tarefa simples. A ferramenta GraphMiner disponibiliza uma interface gráfica que é bastante intuitiva e na qual o usuário pode navegar, extrair informações e transformá-las em conhecimento útil.

Esta abordagem segue o preceito de CHUNG [W. Chung and Jr, 2002], segundo quem a visualização de dados é o processo de apresentação de informação, codificada em um formato visual que pode ser percebido por olhos humanos. A visualização de informações em bancos de dados pode revelar relacionamentos ocultos e implícitos entre



**Figura 5.5.** Grafo de dados visualizados na GraphMiner

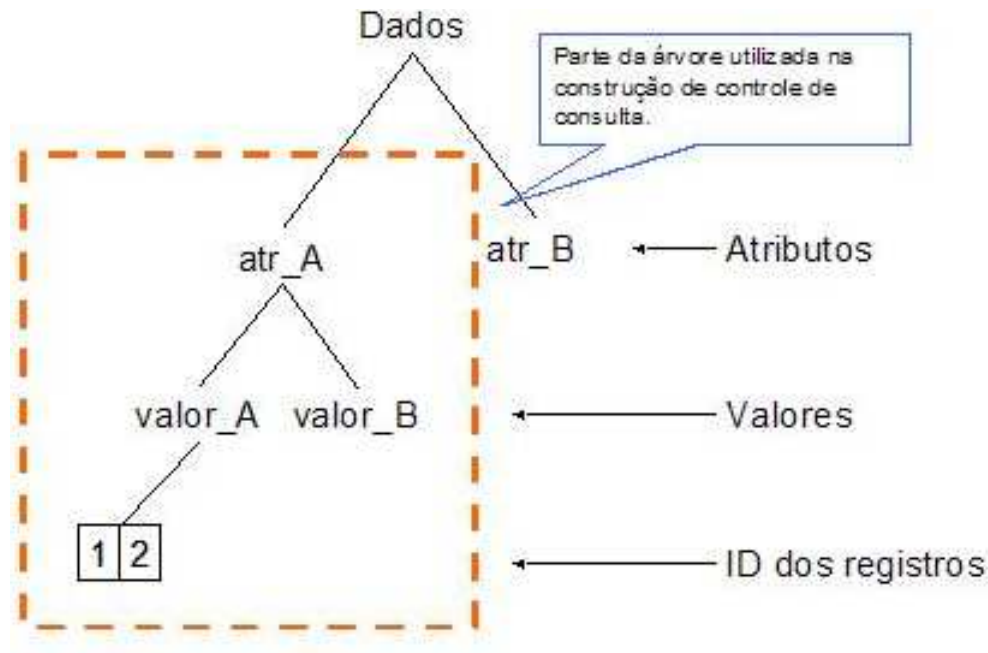
suas diversas entidades. Na GraphMiner, isto é conseguido através da associação dos conjuntos de dados a uma série de atributos visuais.

Cada atributo visual pode ser associado com o valor de um atributo das entidades da base. Desta forma, o usuário da ferramenta GraphMiner pode identificar os dados e suas relações através de atributos visuais que chamam a atenção para mostrar informações importantes.

### 5.3.3 Módulo de Controle de Consulta

Segundo SHNEIDERMAN [Shneiderman, 1996], os controles de consulta são componentes visuais que permitem a execução de operações de seleção sobre o conjunto de dados manipulado. Desta forma os controles de consulta permitem a filtragem dos dados a serem mostrados na tela visual.

Para cada atributo de uma das entidades do banco é criado um controle. Associada ao controle, é criada uma estrutura de dados em árvore. Essa estrutura representa os atributos, os valores que estes podem assumir e quais registros estes valores estão referenciando. A Figura 5.6 exibe organização da árvore de dados que é usada para a criação dos controles de consulta.



**Figura 5.6.** Árvore de dados que é usada para a criação dos controles de consulta.

Na ferramenta GraphMiner, o controle de consulta está baseado apenas em barras de intervalo. Neste controle pode-se selecionar uma faixa de valores para o atributo representado. Pode-se pressionar a seta da direita para ajustar o limite superior da seleção, a seta da esquerda para se ajustar o limite inferior da seleção e a barra central para se ajustar os dois limites ao mesmo tempo. A Figura 5.7 exibe o controle de consulta baseado em barras de intervalo.



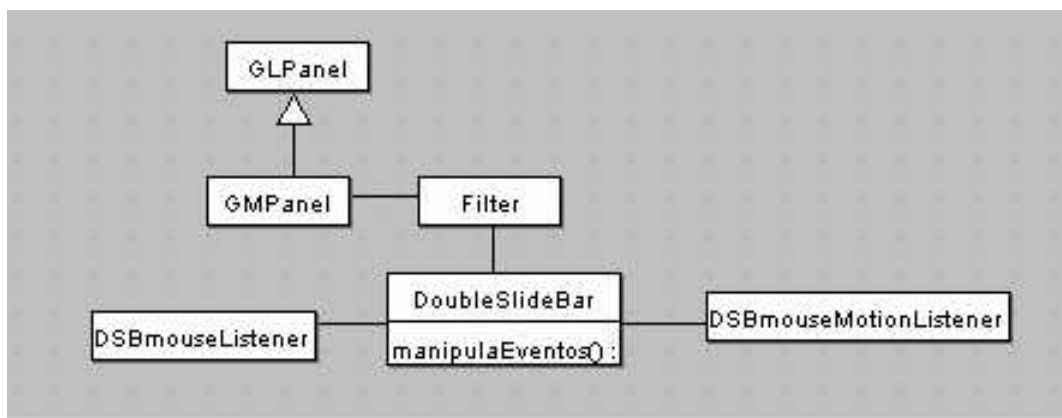
**Figura 5.7.** Barra de intervalo usada como controle de consulta.

As barras de intervalo são implementadas na ferramenta GraphMiner através

das classes *DoubleSliderBar*, *DSBmouseListener*, *DSBmouseMotionListener* e *Filter*. A classe *Filter* é um repositório para objetos do tipo *DoubleSliderBar*. Quando uma entidade pertencente ao conjunto manipulado é selecionada, para cada um dos seus atributos, um objeto do tipo *DoubleSliderBar* é criado e armazenado em um objeto do tipo *Filter*. A classe *DoubleSliderBar* é uma composição de objetos usados para, entre outras tarefas:

- i) Desenhar formas geométricas como retângulos que representam os botões e a barra de rolagem do componente.
- ii) Desenhar setas que rotulam os botões indicando a direção que a barra de rolagem pode ser movimentada.
- iii) Manipular eventos de mouse.

As classes *DSBmouseListener* e *DSBmouseMotionListener* são responsáveis pelos métodos que detectam operações realizadas através do mouse como clicar e arrastar um dos botões de controle da barra. O diagrama das classes que implementam o controle por barras de intervalo na ferramenta GraphMiner e mostrado na Figura 5.8.



**Figura 5.8.** Diagrama das classes que implementam o controle por barras de intervalo.

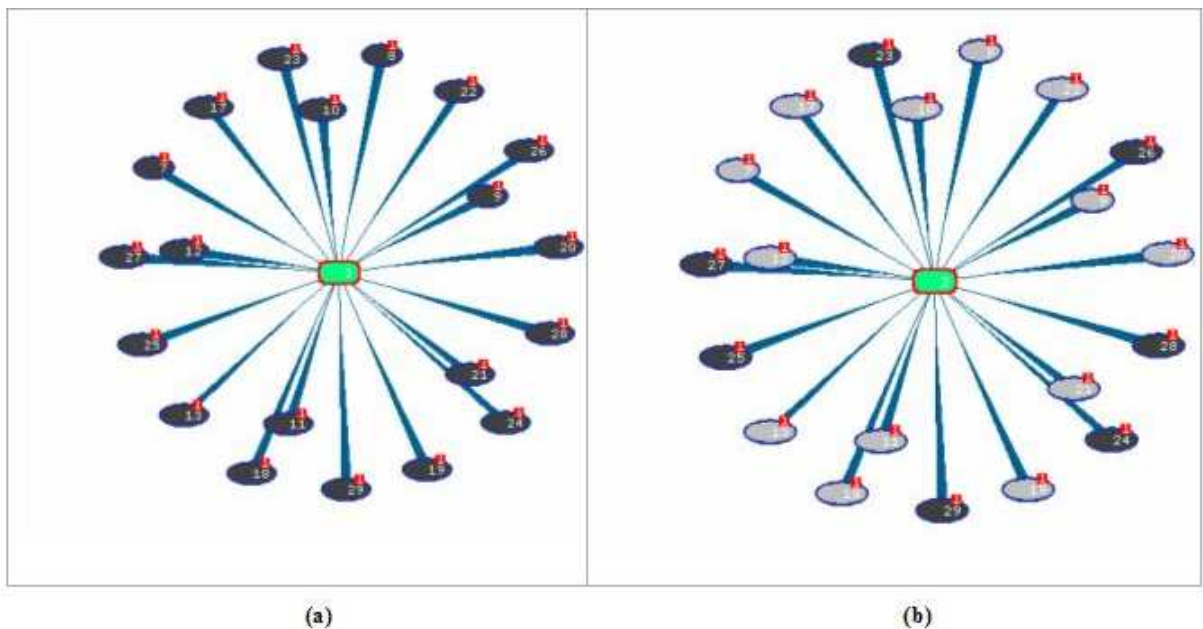
O controle baseado em barras de intervalo torna-se particularmente útil na manipulação de dados numéricos. Existem dois tipos de dados com os quais o GraphMiner trabalha: os dados categóricos e os dados numéricos.

Segundo ALMEIDA [Almeida, 2003], dados numéricos são aqueles que possuem sentido de unidade. Dados categóricos possuem apenas sentido de ordem ou cate-



goria. Por exemplo, o conjunto dos tipos veículos existentes (“Passeio”, “Carga”, “Luxo”) e a capacidade do tanque de combustível de cada um deles (45 litros, 50 litros e 60 Litros) representam dados categóricos e numéricos, respectivamente.

Ao utilizar os controles baseados em barras de intervalo nos dados, os itens que não fazem parte da seleção desaparecem da tela visual quando a opção “Esconder nós após filtragem” está selecionada. Se esta opção não estiver marcada, os nós serão coloridos de forma diferenciada. A Figura 5.9 (a) mostra um grafo cujos nós representam uma lista de valores numéricos que variam de 7 até 29. Esses valores são representados em nós de cor preta e no formato de elipse. Pode-se aplicar o controle de consulta para colorir de cinza os nós cujo valor seja inferior a 23. A Figura 5.9 (b) mostra o resultado da aplicação do controle de consulta baseado em barras de intervalo.



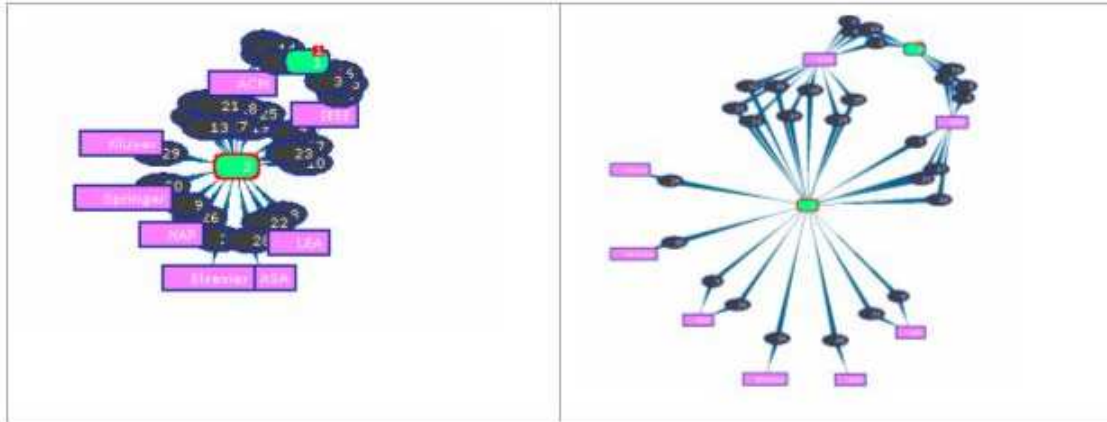
**Figura 5.9.** Grafo antes e depois da utilização dos controles de consulta.

#### 5.3.4 Módulo de Controle de Interação

Com o objetivo de facilitar a exploração do modelo visual a ferramenta Graph-Miner oferece controles que permitem ao usuário interagir com a estrutura visual. Uma série de modificações é possível com o uso desses controles de interação. Os itens a seguir

discutem as funcionalidades e a implementação de cada um deles.

**5.3.4.1 Controle de Aproximação (zoom)** O *zoom* permite aumentar e diminuir as distâncias entre os nós através da modificação do tamanho das arestas. Essa funcionalidade facilita a visualização de informações amenizando problemas como o cruzamento de arestas e a sobreposição de nós conforme exibido na Figura 5.10.



**Figura 5.10.** Grafo antes e depois da utilização dos controles de interação baseado em aproximação.

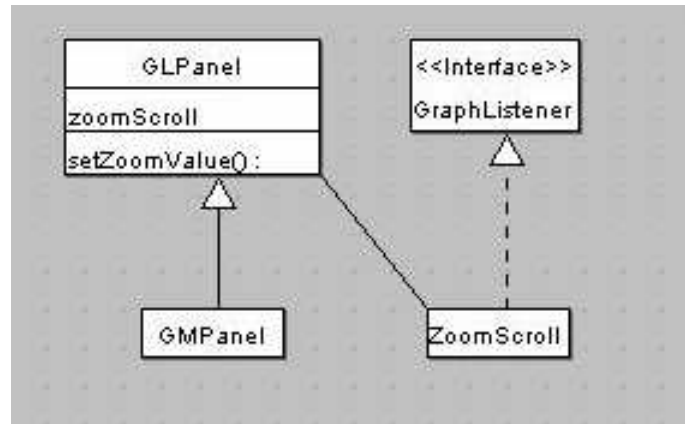
O controle de interação que permite o zoom no grafo é uma das funcionalidades oferecidas pela GraphMiner. Através de uma barra de rolagem denominada *JScrollBar*, um usuário pode modificar o grau de aproximação aplicado ao grafo que pode variar de -100 a +100 unidades. A barra de rolagem da ferramenta GraphMiner usada para configuração do zoom é mostrada na Figura 5.11.



**Figura 5.11.** Barra de rolagem da classe *GMPanel*.

A Figura 5.12 exhibe o diagrama de classes da ferramenta GraphMiner relacionadas ao zoom. A classe *GMPanel* da ferramenta disponibiliza a barra de rolagem usada pelo usuário para modificar o zoom no grafo. Um dos métodos herdados pela classe *GMPanel* da classe *GLPanel*, que faz parte do arcabouço *TouchGraph*, é o *setZoomValue*.

Esse método usa o atributo `zoomScroll` do tipo *ZoomScroll*. A classe *ZoomScroll*, que implementa a interface *GraphListener*, contém os métodos que realizam as operações de redução ou aumento do tamanho das arestas simulando o procedimento de aproximação no grafo.



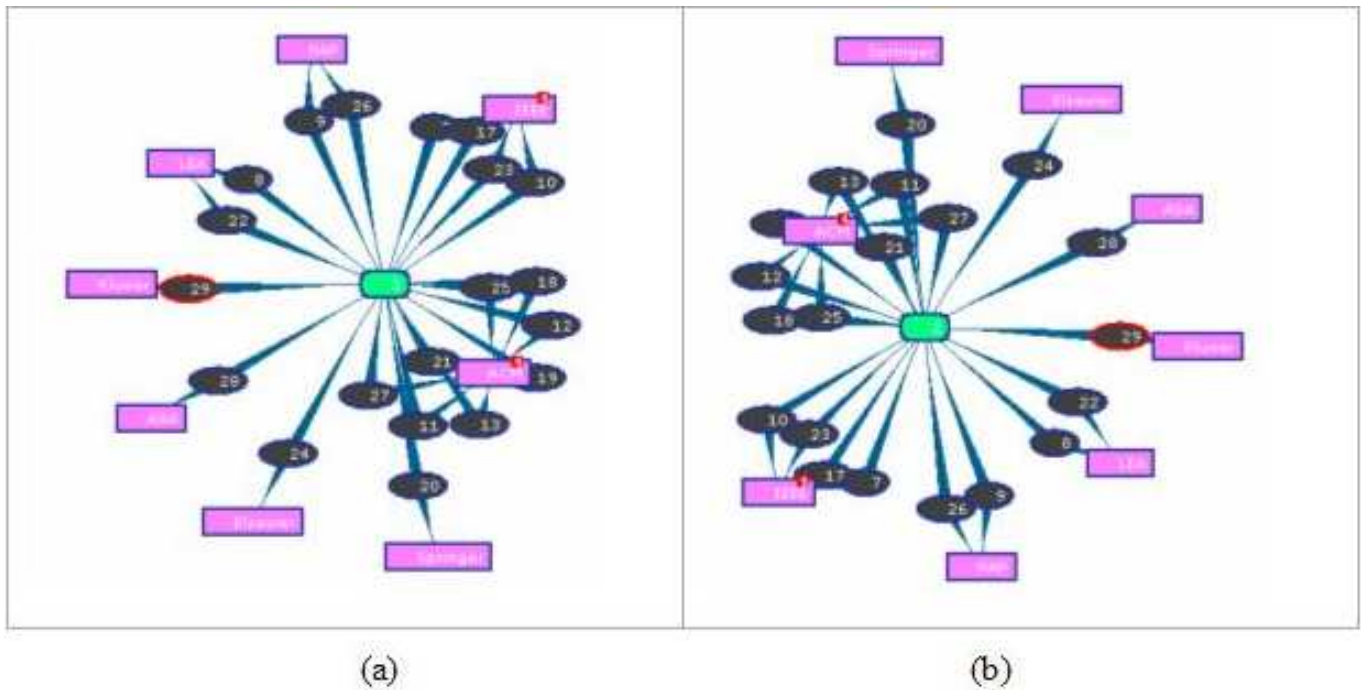
**Figura 5.12.** Diagrama das classes que implementam o controle interação com usuário zoom.

**5.3.4.2 Controle de Rotação** A ferramenta GraphMiner disponibiliza um controle de interação que permite ao usuário girar o grafo em torno do nó centralizado denominado controle de rotação. Através de uma barra de rolagem similar à usada no controle de aproximação pode-se girar modificar as posições dos nós atribuindo valores que vão de 0 a 359 graus. A Figura 5.13 mostra um grafo centralizado no nó de cor verde antes (a) e depois (b) de se aplicar uma rotação de 180 graus.

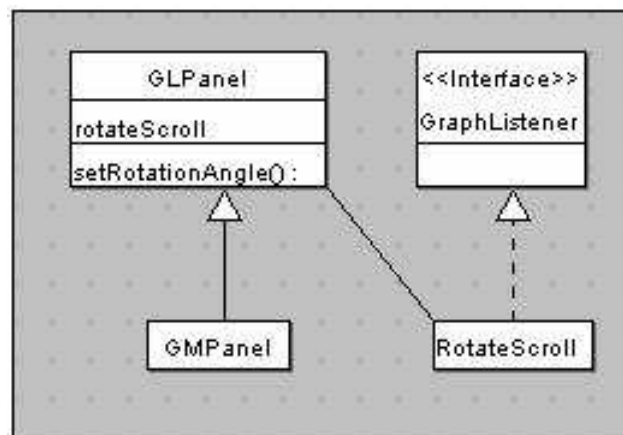
A Figura 5.14 exibe o diagrama de classes da ferramenta GraphMiner com o foco no controle de rotação. Esse controle é mais uma das funcionalidades herdadas do TouchGraph através da classe *RotateScroll*. A classe *RotateScroll*, que como a *ZoomScroll* do controle de aproximação, implementa a interface *GraphListener*, contém os métodos que realizam o procedimento de rotação dos nós e arestas do grafo em torno do nó central.

A classe *GMPanel* herda o método *setRotationAngle* que usa o atributo do tipo *RotateScroll* da classe para modificar o posicionamento dos elementos do grafo. A chamada ao método *setRotationAngle* feita a partir do evento gerado pela barra de

rolagem da mesma forma que acontece com o zoom.



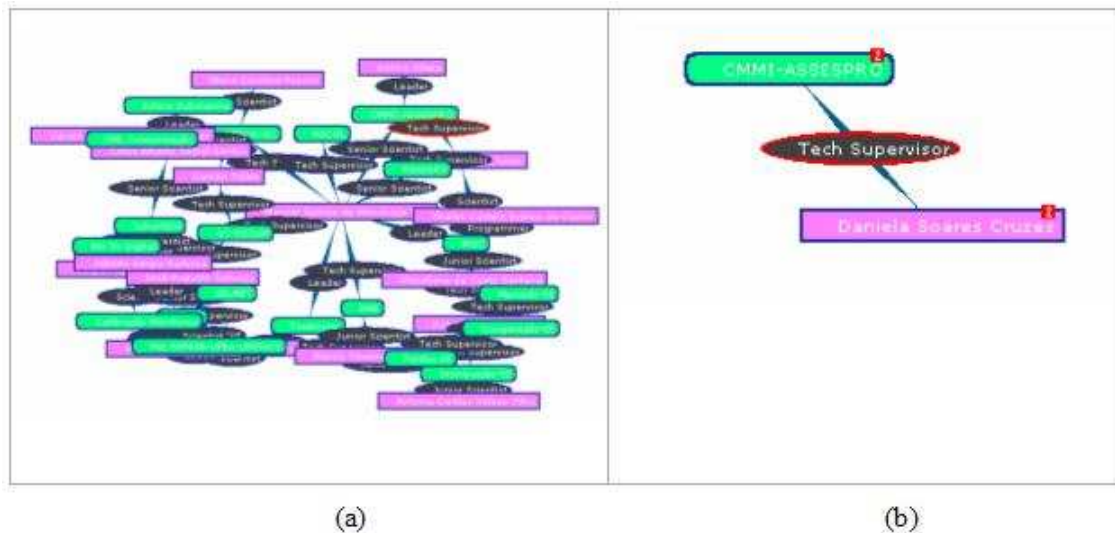
**Figura 5.13.** Grafo antes (a) e depois (b) da utilização dos controles de interação que permite girar o grafo em torno do nó centralizado (de cor verde).



**Figura 5.14.** Diagrama das classes que implementam o controle interação para rotação.

**5.3.4.3 Controle do Raio de Exibição** Outra funcionalidade que segue a mesma linha do controle e aproximação e o de rotação é o controle do raio de exibição. Ele permite controlar a quantidade de nós exibidos na tela usando como parâmetro para mostrar ou ocultar um nó a sua distância do nó selecionado. Nesse caso, a distância

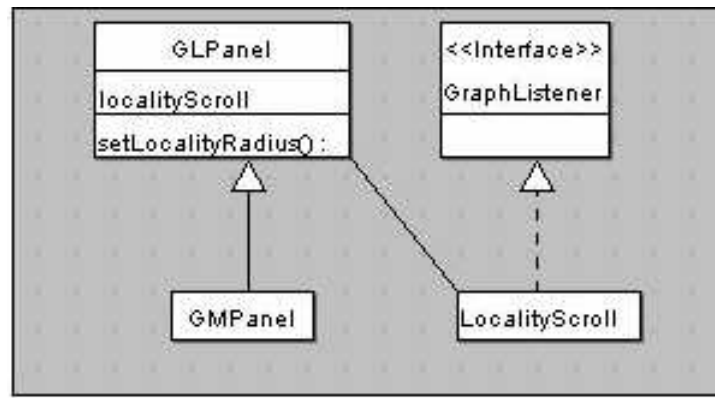
entre dois nós corresponde à quantidade de arestas entre o nó de origem selecionado e nó destino. Quando o nível de visualização do grafo é zero, somente o nó selecionado é mostrado. O controle do raio de exibição só vai até o valor 4, a partir desse valor todos os nós do grafo são exibidos. O número de nós visíveis na tela do computador pode ajudar a melhorar a visibilidade do grafo conforme mostrado na Figura 5.15.



**Figura 5.15.** Grafo com nível de exibição configurado com valor maior que quatro. (b) Grafo com nível de exibição configurado com valor igual a um.

A Figura 5.16 exibe o diagrama de classes da ferramenta GraphMiner com o foco no controle do raio de exibição. A utilização desse controle é feita a partir da classe *GMPanel*. O método *setLocalityRadius* dessa classe usa o atributo *localityScroll* do tipo *LocalityScroll* para definir o nível de visualização do grafo. Esse método é executado quando o usuário usa uma barra de rolagem semelhante àquela do controle de aproximação. A classe *LocalityScroll* que cumpre a interface *GraphListener*, contém os métodos que possibilitam exibir ou ocultar nós do grafo a partir da distância entre dois nós.

**5.3.4.4 Controle de Mapeamentos** A ferramenta disponibiliza ainda os controles de interação que permitem o mapeamento de atributos visuais exibidos no grafo como a cor, o formato e o rótulo dos nós e arestas com qualquer um dos atributos do banco de dados relacional. Esse mapeamento visa tornar intuitiva a identificação dos itens do



**Figura 5.16.** Diagrama das classes que implementam o controle de níveis de exibição do grafo.

grafo facilitando a inspeção visual dos dados realizada pelo usuário.

A escolha do atributo visual é feita através de uma caixa de seleção que contém a lista de atributos de uma das entidades do banco. Cada caixa de seleção está associada a um dos quatro atributos visuais do grafo. O atributo selecionado a partir de uma das caixas, é classificado como categórico ou numérico e, a partir dessa classificação, a modificação de um atributo visual é executada. Por exemplo, associando-se um atributo categórico (Bom, Ruim ou Regular) com o controle de cor do nó, uma cor (ex: Azul, Vermelha ou Preta) é associada a cada nó (Bom-Azul, Regular-Vermelha, Ruim-Preta). As caixas de seleção são mostradas na Figura 5.17.

The image shows a vertical stack of four selection boxes. Each box has a label on the left and a dropdown arrow on the right. The labels are: Node Color, Node Shape, Edge Color, and Node Label.

**Figura 5.17.** Caixas de seleção dos atributos das entidades do banco para mapeamento dos atributos visuais do grafo.

Um outro mapeamento que pode ser realizado é a associação de um ícone a

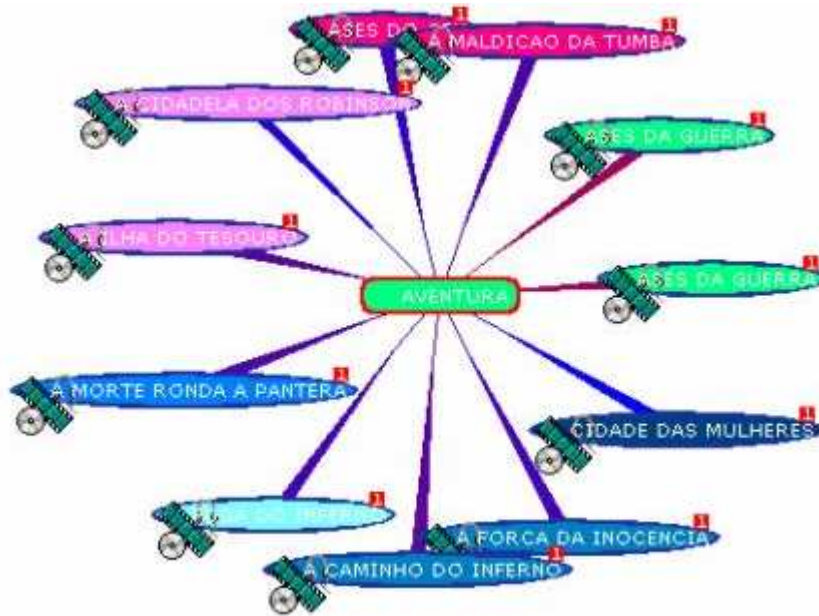
uma das entidades selecionadas. Essa funcionalidade facilita a identificação dos nós que representam as tuplas da entidade selecionada. Por exemplo, a Figura 5.18 exibe um grafo cujos nós exibem mapeamentos de atributos visuais com atributos do banco de dados relacional.

Neste banco, que guarda as informações sobre os filmes de uma locadora de vídeo, existem três entidades: Filme, gênero e produtora. Os nós no formato de elipse e com uma imagem associada são do tipo filme e o nó retangular no centro é do tipo gênero e os nós do tipo produtora estão ocultos pelo controle do raio de exibição mencionado anteriormente. A cor de cada nó do tipo filme está mapeada com atributo do banco de dados produtora. Isso significa que nós com a mesma cor têm a mesma produtora. Pode-se observar pela figura que os filmes “A Ilha do Tesouro” e “A Cidadela dos Robinson” são da mesma produtora pois tem a mesma cor.

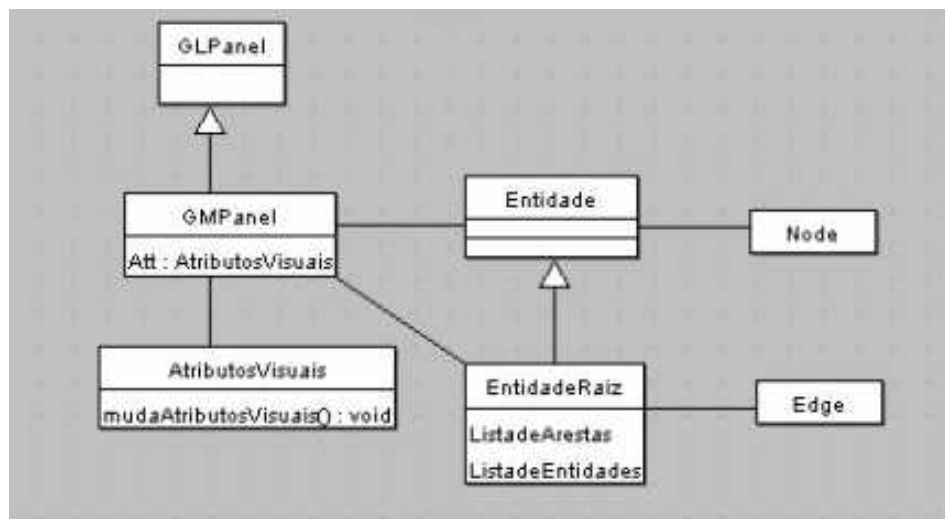
Quando se mapeia o atributo visual cor (de aresta ou nó) a um atributo numérico, uma escala de cores é construída. A escala de cores vai do vermelho ao azul passando pelo marrom ou seja, nós ou arestas cuja cor está próxima da vermelha têm valor próximo do menor valor do atributo no banco e nós ou arestas de cor próxima da azul tem valor próximo do maior valor do atributo no banco. Na Figura 5.18, a cor das arestas que conectam os nós são mapeadas com o atributo “tempo de duração do filme” que é classificado como numérico. Pela cor das arestas pode-se constatar que os filmes “Ases da Guerra” (cor de aresta Marrom e tempo 50 minutos) têm tempo de duração menor que o filme “Cidade das Mulheres” (cor de aresta Azul e tempo 134).

O diagrama de classes com foco nos controles de mapeamento é mostrado na Figura 5.19. Os controles usam métodos do TouchGraph associados com controles visuais desenvolvidos na GraphMiner. A classe AtributosVisuais da ferramenta concentra os métodos que controlam o mapeamento dos atributos visuais do grafo com os atributos das entidades selecionadas para visualização (ver seção 5.3.2). O método mudaAtributosVisuais dessa classe é usado pela classe *GMPanel* para realizar a operação de mapea-

mento escolhida pelo usuário. A classe *GMPanel* tem a responsabilidade de passar como parâmetro para o método a lista de objetos do grafo que irão sofrer modificações em seus atributos visuais. Essa lista está presente nos objetos do tipo Entidade conforme exibido na Figura 5.19.



**Figura 5.18.** Grafo com mapeamentos de atributos visuais.



**Figura 5.19.** Diagrama das classes que implementam o mapeamento dos atributos visuais do grafo.

Detalhes sobre as informações associadas aos nós do grafo são exibidos em uma tabela. Localizados no canto superior direito da ferramenta, estes detalhes são obtidos



selecionando um dos nós. O controle de interação que disponibiliza essa funcionalidade é chamado de detalhes sob demanda. Considere o nó rotulado “Ases da Guerra” na Figura 5.18. Este nó exibe visualmente os atributos “Nome” (Rótulo), “Produtora” (Cor) e “Tempo de Duração” (Cor da Aresta). Os detalhes do registro representado por este nó podem ser vistos na tabela de detalhe conforme mostrado na Figura 5.20.

Attribute	Value
CODIGO	0001A
NOME	ASES DA GUERRA
GENERO	AVENTURA
PRODUTORA	AMERICA
DATCOM	1989-12-12 00:00:00.0
ANOPRO	1998
TEMDUR	50

**Figura 5.20.** Detalhes sobre demanda de um nó do grafo.

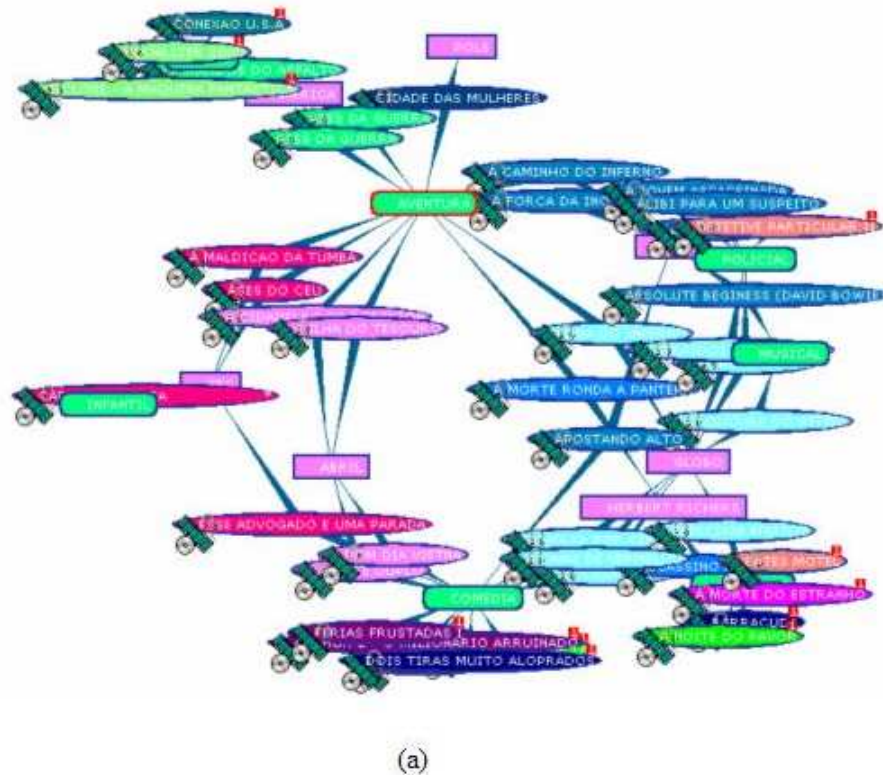
A Figura 5.21 mostra o diagrama de classes responsáveis pelo controle de detalhes sob demanda. As classes do controle de detalhes sob demanda são *TableModel*, *HVScroll* e *Table*. A classe *HVScroll* pertence ao *TouchGraph* e sofreu uma modificação no método *slowScrollToCenter* usado para centralizar um nó no painel. O método agora é usado pela classe *GMPanel* tanto para centralizar um nó quanto para mostrar, através de um objeto do tipo *TableModel*, os detalhes do nó. A classe *TableModel* é apenas o repositório de dados passado como parâmetro para classe *Table* que é a classe responsável pela exibição.



**Figura 5.21.** Diagrama das classes do controle de detalhes sob demanda.

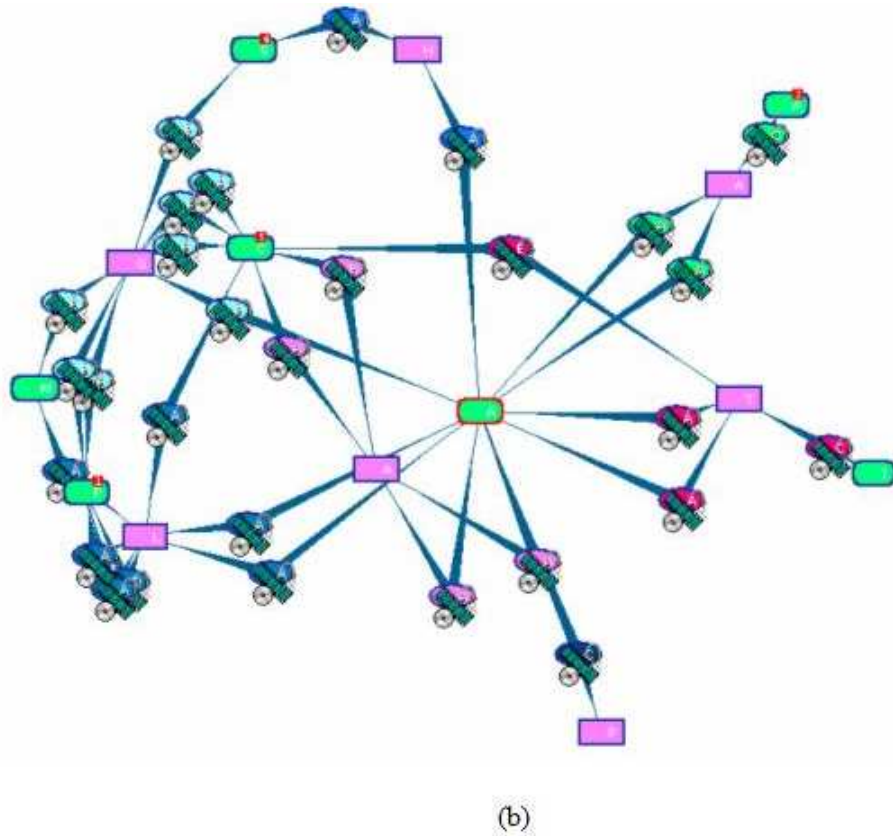
**5.3.4.5 Outros Controles** Existe ainda um controle de interação que permite buscar uma informação usando uma palavra-chave. A busca é realizada através de uma caixa

de texto onde o usuário pode digitar o objeto da consulta e de um botão que aciona o procedimento de busca. O primeiro nó do grafo cujo rótulo seja igual ou pelo menos contenha a palavra-chave escolhida na busca, será selecionado e centralizado na tela de exibição. Esse controle facilita a localização de informações na cena visual. A classe *GM-Panel* é responsável por esse controle. O método *findNodeByLabel* da classe *TGPanel* é usado para encontrar nós no grafo.



**Figura 5.22.** (a) Grafo exibindo o rótulo completo.

Um outro controle chamado de *minView* (*MinimalView*) permite que seja exibido apenas o primeiro caractere do rótulo. Esse controle diminui o tamanho dos nós melhorando a visibilidade do grafo. A Figura 5.22 (a) mostra o mesmo grafo da Figura 5.18 mas com o nível de exibição configurado para 4. Percebe-se que o número elevado de nós exibidos dificulta a identificação dos mesmos. Aplicando-se o *minView* pode-se obter mais facilmente a identificação dos nós do tipo filme pela imagem associada a cada um deles conforme visto na Figura 5.23 (b).



**Figura 5.23.** (b) Grafo exibindo apenas o primeiro caractere.

Todos os controles de interação disponíveis na GraphMiner têm o objetivo de apoiar o processo de descoberta de informações implícitas em bancos de dados relacionais. Esta seção detalhou a implementação de cada um desses controles. A próxima seção discute as características básicas da GraphMiner frente ao que foi discutido sobre outras ferramentas no capítulo 3.

#### 5.4 COMPARANDO A FERRAMENTA GRAPHMINER COM OUTRAS FERRAMENTAS

A GraphMiner é uma ferramenta voltada à **visualização** de relações entre informações desenvolvida como parte desta dissertação que segue os princípios de uma ferramenta para mineração visual de dados (ver seção 2.3). Ela usa como metáfora visual um grafo e tem como objetivo montar uma representação visual dos relacionamentos de dados contidos em **bancos de dados relacionais** permitindo a descoberta de informações potencialmente úteis e inéditas até então.

A partir de mecanismos de acesso a uma base de dados, um grafo é construído e apresentado de forma gráfica e navegacional. Na construção da cena visual, os registros das diversas entidades são representados como nós e as relações entre eles como arestas.

Seguindo a tendência da maioria das ferramentas da área de visualização baseada em grafos, a GraphMiner foi construída usando os algoritmos de **força-dirigida**. Esses algoritmos já estão implementados no arcabouço TouchGraph que é usado como base da construção da ferramenta.

As maiores vantagens da GraphMiner em relação a outras ferramentas estão na disposição intuitiva dos grafos conseguida através do arcabouço TouchGraph e nos controles de interação com usuário. A GraphMiner estende os **controles de visualização** do TouchGraph incluindo rotação do grafo, navegação sobre a estrutura visual e zoom, além de implementar outros controles que permitem a modificação de atributos visual baseados em dados reais. Isso facilita a interação do usuário durante uma inspeção visual. Através desses controles é possível:

- i) Filtrar (ocultar, exibir ou mudar a cor) os nós que representam as informações exibidas.
- ii) Atribuir qualquer um dos valores dos atributos de uma tabela do banco de dados aos rótulos dos nós que representam seus registros.
- iii) Encontrar uma informação no grafo de forma simples através da busca por palavra chave.
- iv) Associar uma imagem (ícon) a cada um dos nós de uma determinada entidade do banco.
- v) Mudar a entidade raiz do grafo sem a necessidade de acessar o banco novamente.
- vi) Exibir sob demanda os valores dos atributos de cada um dos nós do grafo.

- vii) Associar a cor dos nós e das arestas do grafo com atributos categóricos (“Bom”, “Ruim” e etc.) ou numéricos (10, 11, 12 e etc.) do banco de dados.
- viii) Associar o formato dos nós com atributos categóricos ou numéricos do banco de dados.

## 5.5 CONSIDERAÇÕES FINAIS

Neste capítulo discutiu-se a implementação da ferramenta GraphMiner, originada a partir das idéias apresentadas nesta dissertação. Sua principal contribuição reside em visualizar as relações dos dados em bancos relacionais no intuito de facilitar a descoberta de informações potencialmente úteis e inéditas até então. Neste contexto, foram mostrados alguns dos principais artefatos produzidos durante o seu processo de desenvolvimento, visando o melhor entendimento das funcionalidades disponíveis na ferramenta. Este capítulo finaliza caracterizando a ferramenta frente aos conceitos e ferramentas discutidos nos capítulos 2, 3 e 4.

No próximo capítulo são apresentados dois exemplos de uso desenvolvidos com o intuito de subsidiar a avaliação da aplicabilidade da ferramenta.

# EXEMPLOS DE USO DA FERRAMENTA GRAPHMINER

Para avaliar a versão aqui apresentada da ferramenta GraphMiner, foram realizados dois exemplos de uso, envolvendo uma base de dados dos projetos realizados no Núcleo Interdepartamental de Pesquisas em Redes de Computadores (NUPERC) e um relatório sobre o ranking publicações em visualização de informação (InfoVis).

Este capítulo aborda as características operacionais da ferramenta, bem como a experiência de aplicação da mesma nos contextos citados. Para cada exemplo de uso são descritos: o domínio da aplicação, os dados envolvidos e o processo de desenvolvimento. Ao final deste capítulo é feita uma avaliação geral da ferramenta GraphMiner, apontando seus pontos fortes e fracos encontrados durante o processo.

### 6.1 EXEMPLO DE USO NUPERC

Nesta seção será descrito o desenvolvimento do exemplo de uso que envolveu informações de uma base de projetos desenvolvidos no Núcleo Interdepartamental de Pesquisas em Redes de Computadores da Universidade Salvador - UNIFACS.

#### 6.1.1 Caracterização da Organização

O NUPERC é um grupo de pesquisa associado aos Departamentos de Engenharia e Arquitetura (DEAR) e de Ciências Exatas (DCEX) da UNIFACS. Esse grupo foi concebido para operar integrado ao Mestrado Profissional em Redes de Computadores da Universidade Salvador e realiza pesquisas relacionadas com:

- i) Avaliação de desempenho de soluções de redes de alta velocidade;

- ii) Estudo e a avaliação de mecanismos para garantia de QoS (Qualidade de Serviço);
- iii) Aplicações multimídia sob redes;
- iv) Estudo da arquitetura e protocolos de redes ópticas;
- v) Controle de tráfego e congestionamento em redes de alta velocidade;
- vi) Desenvolvimento de novas tecnologias e aplicações para a Internet;
- vii) Mineração de Dados.

Os trabalhos que são realizados no NUPERC incluem, o desenvolvimento de projetos de Pesquisa e Desenvolvimento, a realização de consultorias, a divulgação de resultados em eventos e periódicos, a realização de cursos de treinamento, a participação de seus pesquisadores em cursos regulares de graduação e pós-graduação, etc. Atualmente o grupo desenvolve pesquisas com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Financiadora de Estudos e Projetos (FINEP), do Plano Nacional de Ciência e Tecnologia do Setor Petróleo e Gás Natural (CT-PETRO), da Agência Nacional do Petróleo (ANP) e da Agência Nacional de Energia Elétrica (ANEEL), entre outras.

### **6.1.2 Domínio de Aplicação**

As informações envolvidas neste exemplo de uso são provenientes das bases de dados de projetos desenvolvidos no núcleo em diversas áreas. Foram selecionados 20 projetos ligados a 8 áreas de conhecimento (redes, mineração de dados, Internet, hardware, engenharia de software, petróleo, energia e gestão do conhecimento). Nesses projetos, estão envolvidas 17 pessoas que exercem diferentes papéis a depender do projeto que estão ligadas.

Inicialmente foi feita uma coleta das informações existentes no núcleo que poderiam compor a base de dados usada pela ferramenta. Na próxima seção são de-

scritos os processos de visualização desses dados, bem como o conteúdo das informações utilizadas neste exemplo de uso.

### 6.1.3 Sobre os Dados

Os dados coletados para este exemplo de uso foram extraídos de diversas fontes e armazenados em uma base local acessada pela ferramenta. A etapa de pré-processamento fez a transformação dos dados para adequá-los à estrutura da base. Esta etapa englobou as operações de: concatenação e quebra de dados, limpeza de caracteres especiais, exclusão de atributos não relevantes e redução de escala para a transformação de campos numéricos ou alfanuméricos em campos categóricos.

### 6.1.4 Desenvolvimento do exemplo de uso

O desenvolvimento do exemplo de uso se iniciou com a coleta de todas as informações relacionadas aos projetos envolvidos no estudo de caso. Depois foram cadastradas no banco relacional as pessoas que compõem o quadro de indivíduos envolvidos nestes projetos, conforme ilustrado na Figura 6.1.



	PersonName	Title	Graduation	Function	Person_id
+	Manoel Gomes de Mendonça Neto	Ph.D.	Computing	Professor	1
+	Celso Alberto Saibel Santos	Ph.D.	Computing	Professor	2
+	Joberto Sergio Barbosa	Ph.D.	Computing	Professor	3
+	José Augusto Suruagy	Ph.D.	Computing	Professor	4
+	William Ferreira Giozza	Ph.D.	Computing	Professor	5
+	Daniela Soares Cruzes	Ph.D. Candidate	Computing	Researcher III	6
+	Márcio Oliveira Almeida	M.Sc.	Computing	Researcher II	7
+	Roberto Dantas Pinho	M.Sc.	Engineering	Researcher III	8
+	Sérgio Figueira de Brito	Ph.D. Candidate	Engineering	Researcher III	9
+	Camilo Telles	B.Sc.	Computing	Researcher II	10
+	Cássio Vinicius Serafim Prazeres	M.Sc.	Computing	Researcher II	11
+	Christiane da Costa Santana	B.Sc.	Computing	Researcher II	12
+	Leobino Sampaio	B.Sc.	Computing	Researcher II	13
+	Maria Carolina Passos	M.Sc.	Computing	Researcher II	14
+	Antonio Dantas Veloso Filho	B.Sc.	Computing	Researcher I	15
+	Thales Castelo Branco de Castro	B.Sc.	Computing	Researcher I	16
+	Karina Villela	Ph.D.	Computing	Professor	17

Registro: 1 de 17

Figura 6.1. Tabela de Pessoas do banco de dados de projetos do NUPERC.



O cadastramento das pessoas que envolveu a coleta das informações de nome, titulação, tipo de graduação, função exercida no projeto e número de identificação. O atributo titulação descreve o título de maior importância da pessoa cadastrada e contempla as categorias “Doutor”, “Doutorando”, “Mestre” e “Bacharel”. O atributo função também é categórico e pode assumir os valores “Professor”, “Pesquisador I”, “Pesquisador II” e “Pesquisador III”.

Após esse cadastramento, foi possível a realizar o cadastramento dos projetos selecionados para este exemplo. A Figura 6.2 mostra a tabela de projetos usada neste exemplo de uso.



ProjectName	KnowledgeArea	GrantAmount	Proj_id
SELMET	Oil	C	1
RECOL	Oil	B	2
EMS	Data Mining	A	3
XML Comunicaç	Internet	B	4
Calibração Moni	Hardware	A	5
Treeminer	Data Mining	A	6
MQC	Oil	D	7
READERS	Software Engineering	A	9
Esfera Submarin	Oil	E	10
RH TV Digital	Computer Networks	C	11
InfraVIDA	Computer Networks	C	12
IQoM	Computer Networks	B	13
GC-RECAM	Oil	C	14
Perdas-EE	Energy	A	15
Coogeração-EE	Energy	A	16
Distribuição-EE	Energy	A	17
Mercado-EE	Energy	A	18
Net FAPESB-Uf	Computer Networks	D	19
UNITECH-GC	Knowledge Management	B	20
CMMI-ASSESPI	Software Engineering	C	21

**Figura 6.2.** Tabela de projetos usada no exemplo de uso do NUPERC.

A tabela de projetos é composta pelos atributos: Nome do projeto, área de conhecimento, valor financeiro hipotético do projeto e número de identificação. O atributo valor que era inicialmente numérico foi convertido para categórico. Os valores numéricos foram divididos igualmente e quatro categorias “A”, “B”, “C”, “D” e “E” sendo que “A”

representa os projetos de menor valor e “E” o de maior valor. As áreas de conhecimento que anteriormente tinham descrições extensas foram classificadas nas oito áreas observadas na seção 6.1.2.

Uma vez que os projetos e as pessoas foram cadastrados, definiu-se que papéis são exercidos pelas pessoas nos projetos selecionados. Isso foi feito construindo-se uma relação entre as duas tabelas já criadas. A Figura 6.3 mostra a relação usada no exemplo de uso do NUPERC.

Proj_Pessoaid	fk_Projeto	Role	fk_Pessoa
34	7	Junior Scientist	12
22	3	Junior Scientist	7
24	15	Junior Scientist	7
33	4	Junior Scientist	11
41	17	Junior Scientist	15
13	1	Leader	4
17	5	Leader	5
3	6	Leader	1
4	7	Leader	1
42	19	Leader	5
11	13	Leader	3
51	21	Leader	17
37	11	Leader	9
9	10	Leader	2
39	7	Programmer	16
32	13	Scientist	13
20	7	Scientist	6

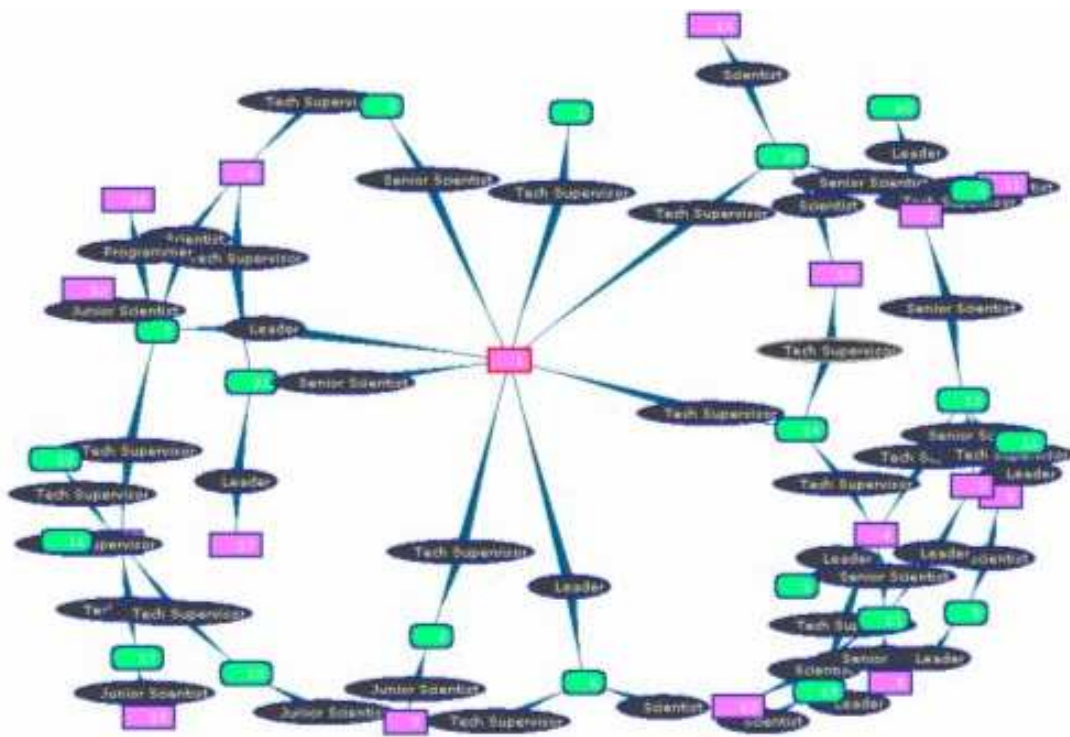
**Figura 6.3.** Relação utilizada no exemplo de uso do NUPERC.

A partir da análise das funcionalidades de cada uma das pessoas nos projetos, foram definidos seis papéis contemplados no atributo papel da relação. Esse atributo pode assumir os valores “Cientista”, “Cientista Júnior”, “Cientista Sênior”, “Líder”, “Programador” e “Supervisor Técnico”.

Com todas as informações necessárias já presentes na base de dados local da ferramenta foi possível realizar a visualização dos dados e realizar consultas sobre eles. As consultas foram elaboradas com o objetivo de verificar se a GraphMiner aos requisitos

funcionais estabelecidos para a ferramenta na seção 5.1 desta dissertação.

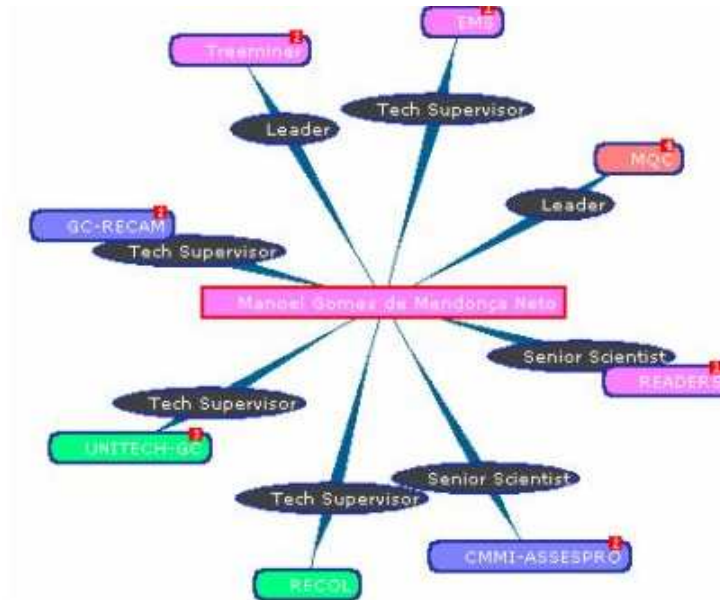
No grafo mostrado na Figura 6.4 estão representados os dados da entidade projeto e as relações dessa com as entidades Pessoas e Papéis. Isto atende ao requisito 1.1, que é mostrar as relações existentes entre os dados através do grafo. Os dados de projetos são representados com nós de cor verde rotulados com o atributo número do projeto cujo formato se assemelha a um retângulo de bordas arredondadas. Os dados das pessoas são representados com nós retangulares de cor rosa rotulados com seu número de identificação e seus papéis são nós de cor preta com formato de uma elipse rotulados com seu atributo papel.



**Figura 6.4.** Grafo dos dados de projetos da base relacional do NUPERC.

A partir da representação visual destes dados, pode-se modificar qualquer um dos atributos visuais do grafo no intuito de facilitar a extração de informações. Por exemplo, a cor dos nós da entidade projeto pode ser definida através do atributo categórico valor e os rótulos pelo atributo nome-projeto. Esse exemplo pode ser observado na Figura 6.5, que mostra o mesmo grafo exibido na Figura 6.4 com o nível de exibição configurado para o valor dois e com as modificações realizadas no exemplo. Os nós de projeto de

cor rosa tem valor enquadrado na categoria “A”, os de cor verde na categoria “B”, os de cor azul na categoria “C” e os de cor vermelha na categoria “D”. Esta associação entre atributos reais e visuais cumpre os requisitos funcionais do item 1.2.



**Figura 6.5.** Grafo com atributos visuais mapeados com atributos do banco.

Além dos mapeamentos citados, pode-se ainda realizar filtragem nos dados exibidos. Esta filtragem atende ao requisito funcional especificado no item 2.2. Usando o exemplo anterior pode-se colorir de cinza os nós cujo atributo valor seja diferente de “A”. Isto pode ser constatado na Figura 6.6.

Durante o desenvolvimento do exemplo de uso foram realizadas diversas outras consultas com o intuito de identificar e organizar os dados dos projetos existentes no NUPERC. O maior interesse do núcleo é promover a melhoria do desempenho dos seus pesquisadores nos projetos na distribuição correta das atividades nos contextos apropriados. Desta forma, um aplicativo que faça a representação visual dos dados relativos aos projetos, a partir das fontes de informação presentes no núcleo, pode se tornar uma ferramenta importante de apoio gerencial.

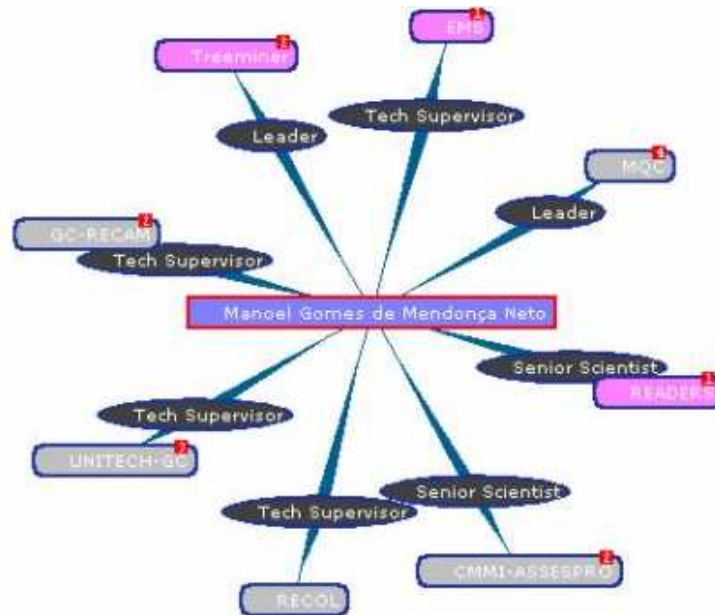


Figura 6.6. Grafo com os nós de projeto filtrados pelo atributo valor.

## 6.2 EXEMPLO DE USO INFOVIS

Nesta seção será descrito o exemplo que envolve informações do ranking de publicações na área de visualização de informação feito por CONKLIN [N.Conklin, 2004].

### 6.2.1 Caracterização do Relatório

Segundo CONKLIN [N.Conklin, 2004], a escolha dos atributos do ranking levou em conta publicações que relatam técnicas de projetos, ferramentas e pesquisas científicas na área de InfoVis. O ranking inclui jornais e anais de conferências. Os atributos usados no ranking são: ranking geral, ranking por jornal, ranking por conferência, frequência de referência, título da publicação e o seu editor.

Os rankings foram gerados a partir de opiniões peritas pois InfoVis é uma área muito nova dentro da ciência da computação. A frequência média de referência foi calculada a partir do número de referências que um conjunto de artigos importantes da área faziam a artigos publicados nos veículos sendo considerados.

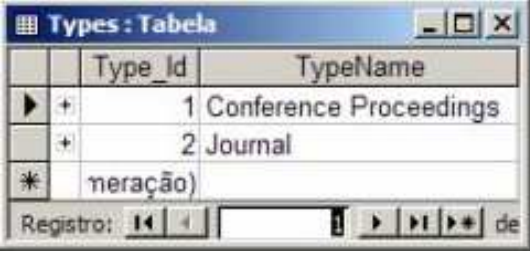
### 6.2.2 Domínio de Aplicação

Este exemplo de uso envolve as informações de cerca de 14 publicações da área de visualização de informações. Esta amostra foi selecionada de publicações inclui jornais e anais de conferência de 9 editores.

Inicialmente foi feita uma análise das informações existentes no relatório. A partir desta análise os dados foram selecionados para posteriormente serem digitados na base de dados relacionais acessada pela ferramenta. Na próxima seção são descritos os processos de seleção e de entrada de dados das informações na base.

### 6.2.3 Desenvolvimento do Estudo de Caso

O desenvolvimento do estudo de caso se iniciou com o cadastramento das informações do relatório na base de dados. Inicialmente foram cadastrados no banco relacional, os tipos de documentos enquadrados no relatório do ranking de publicações. Isso foi feito através da criação da entidade “Tipo” no banco. Essa entidade tem dois atributos: o número de identificação do tipo (numérico) e a sua descrição (categórico) conforme tabela mostrada na Figura 6.7.

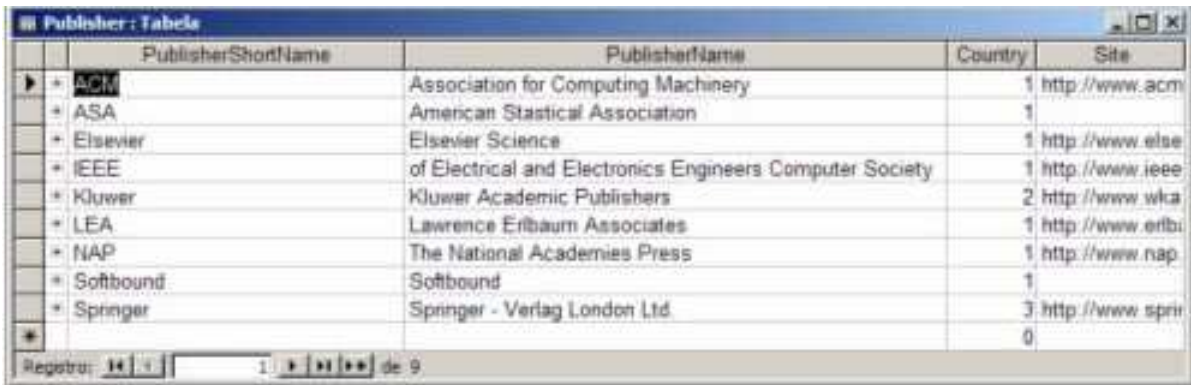


Type_Id	TypeName
1	Conference Proceedings
2	Journal

**Figura 6.7.** Entidade “tipo” usado no exemplo de uso InfoVis.

Após esse cadastramento, passou-se ao cadastro dos institutos de pesquisa onde as publicações foram publicadas. Esse cadastro contemplou os atributos: sigla (categórico), nome (categórico), país e homepage (categórico). A Figura 6.8 mostra a tabela de editores usada neste exemplo de uso. O atributo país da entidade mencionada é um atributo numérico cujo valor é referenciado na entidade país. A entidade país contém os 3 países de origem dos institutos de pesquisa.

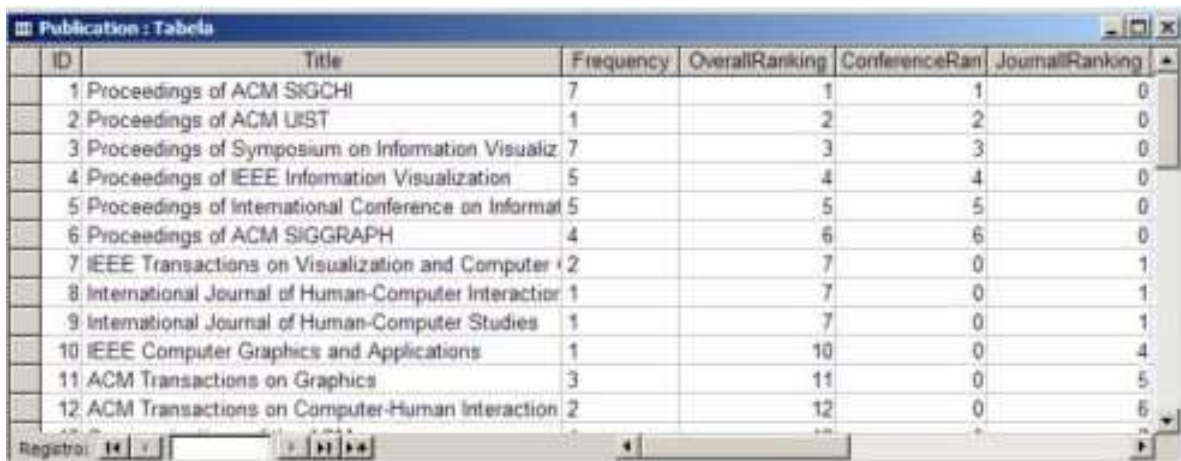




	PublisherShortName	PublisherName	Country	Site
▶ *	ACM	Association for Computing Machinery	1	http://www.acm
▶ *	ASA	American Stastical Association	1	
▶ *	Elsevier	Elsevier Science	1	http://www.else
▶ *	IEEE	of Electrical and Electronics Engineers Computer Society	1	http://www.ieee
▶ *	Kluwer	Kluwer Academic Publishers	2	http://www.wka
▶ *	LEA	Lawrence Erlbaum Associates	1	http://www.erlbi
▶ *	NAP	The National Academies Press	1	http://www.nap
▶ *	Softbound	Softbound	1	
▶ *	Springer	Springer - Verlag London Ltd	3	http://www.spri
			0	

**Figura 6.8.** Tabela “institutos de pesquisa” usada neste exemplo de uso.

Uma vez que os institutos de pesquisa e os tipos de publicação foram cadastrados passou-se a definir que publicações pertenciam a qual instituto e, qual era a categoria que essa publicação se enquadrava (“Jornal” ou “anais de conferência”). Isso foi feito construindo-se uma relação entre as duas tabelas já criadas. A Figura 6.9 mostra a relação “Publicações” usada no exemplo de uso do InfoVis.

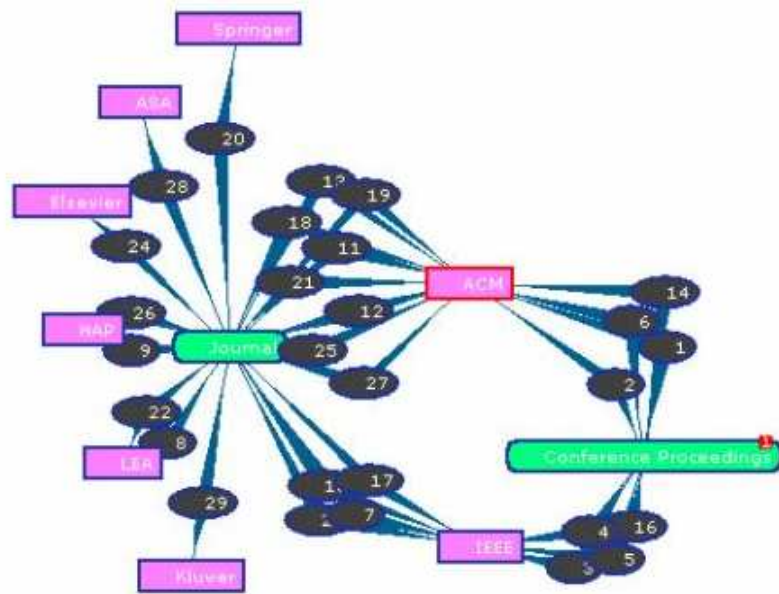


ID	Title	Frequency	OverallRanking	ConferenceRan	JournalRanking
1	Proceedings of ACM SIGCHI	7	1	1	0
2	Proceedings of ACM UIST	1	2	2	0
3	Proceedings of Symposium on Information Visualiz	7	3	3	0
4	Proceedings of IEEE Information Visualization	5	4	4	0
5	Proceedings of Intemational Conference on Informal	5	5	5	0
6	Proceedings of ACM SIGGRAPH	4	6	6	0
7	IEEE Transactions on Visualization and Computer	2	7	0	1
8	International Journal of Human-Computer Interaction	1	7	0	1
9	Intemational Journal of Human-Computer Studies	1	7	0	1
10	IEEE Computer Graphics and Applications	1	10	0	4
11	ACM Transactions on Graphics	3	11	0	5
12	ACM Transactions on Computer-Human Interaction	2	12	0	5

**Figura 6.9.** Relação “Publicações” usada no exemplo de uso InfoVis.

No grafo mostrado na Figura 6.10 estão representados os dados da entidade publicação e as relações dessa com as entidades institutos e tipos. Isto atende aos requisitos funcionais especificados nos itens 1.1 e 2.1. Os dados das publicações são representados com nós de cor preta rotulados com o atributo número da publicação no formato de elipse. Os dados dos institutos são representados com nós retangulares de cor rosa rotulados com sua sigla e o tipo das publicações são nós cor de rosa cujo formato se assemelha a um retângulo de bordas arredondadas. A partir da representação visual dos dados, pode-se

verificar que dois institutos de pesquisa, o IEEE e ACM concentram a maior parte das publicações.



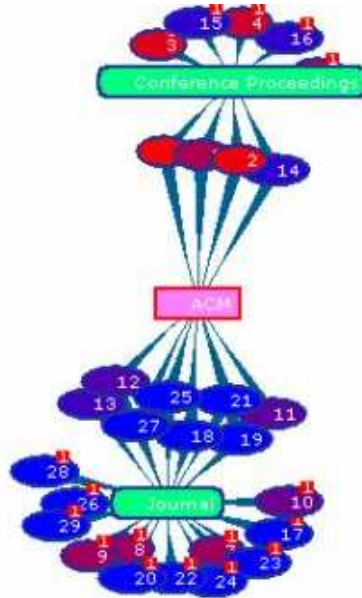
**Figura 6.10.** Grafo dos dados de publicações do relatório InfoVis.

Aplicando-se o mapeamento de cor do nó para o atributo ranking geral do banco pode-se constatar que as publicações do tipo “jornal” estão mais bem classificadas do que os anais de conferência. Essa constatação é feita pela escala de cores aplicadas aos nós cujo atributo visual cor é mapeado com atributos numéricos do banco e atende ao requisito funcional especificado no item 2.1. A escala de cores vai do vermelho ao azul passando pelo marrom ou seja, nós cuja cor está próxima da vermelha têm ranking geral próximo de 1 (menor valor do atributo no banco) e nós de cor próxima da azul ranking geral próximo de 17 (maior valor do atributo no banco). A Figura 6.11 exibe o grafo que atesta a constatação citada.

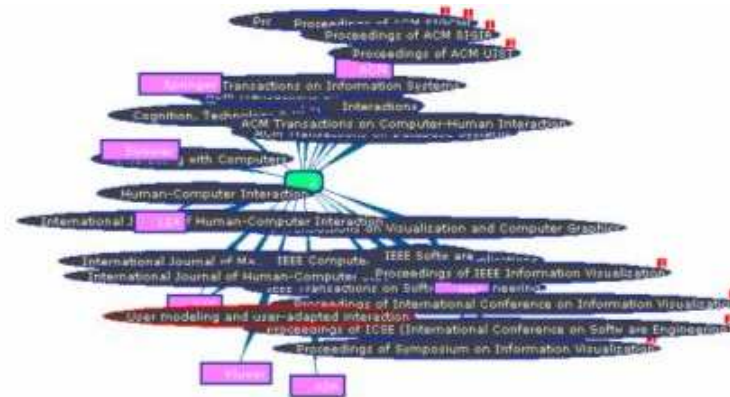
Pode-se ainda mapear o rótulo dos nós de publicações com o atributo título do banco. Esse mapeamento resulta em uma visualização precária como pode ser observado na Figura 6.12. No intuito de atenuar esse problema pode-se aplicar o controle de interação minView (ver 5.3.5) e usar o detalhamento sobre demanda para obter informações específicas do nó. Isto atende aos requisitos funcionais especificados nos itens 2.2 e 2.4.



Durante o desenvolvimento deste exemplo de uso foram criadas diversas outras consultas com o objetivo de identificar, classificar e organizar os dados das publicações contidas no relatório do ranking de publicações na área de InfoVis. Desta forma a visualização de informação com a GraphMiner torna-se um instrumento para apoiar a análise destas publicações e suas relações.



**Figura 6.11.** Grafo com cor do nó mapeada para um atributo numérico do banco.



**Figura 6.12.** Grafo com rótulo dos nós de publicações mapeados para o título.

### 6.3 PONTOS FORTES E FRACOS OBSERVADOS NA GRAPHMINER

A possibilidade de mapear diversos atributos visuais com os dados diferencia a GraphMiner das demais ferramentas vistas. Além disso, um usuário da ferramenta pode

interagir com a cena produzida através de diversos controles de interação modificando os atributos visuais e navegando sobre os dados. Isso é feito com intuito de aumentar a percepção sobre dados visualizados.

Um ponto positivo da GraphMiner é o componente de acesso a dados. Esse componente permite o acesso de forma transparente a qualquer banco de dados relacional, sem necessidade re-programar a ferramenta. Outro fato positivo é mapeamento dos dados armazenados em um banco de dados relacional para grafos, que é feito de forma automática.

Uma das maiores limitações da ferramenta GraphMiner é a quantidade de dados usados no processo de mineração visual. Nos dois exemplos de uso realizados não foram utilizados mais do que 500 registros de dados. As tentativas que superaram esse patamar apresentaram uma visualização precária com muita sobreposição de nós e arestas.

A utilização do controle do raio de exibição ameniza a sobreposição dos elementos do grafo. Porém, não resolve o problema da quantidade de memória insuficiente para comportar a quantidade nós criados para a representação visual quando o número de registros é muito alto.

A questão da quantidade insuficiente de memória pode ser resolvida com a utilização de uma estrutura de dados que permita o armazenamento intermediário (Buffer) entre os dados que estão no disco e os exibidos na tela do computador. Ao contrário do que acontece hoje, os dados seriam carregados sob demanda para estrutura intermediária na memória e, a partir dessa, seriam exibidos na tela. Esta solução é um item que será implementado futuramente na ferramenta. É importante ressaltar também que existe uma limitação humana para a quantidade de informações exibidas na tela. Mesmo que fosse possível exibir uma grande quantidade de nós e arestas sem comprometer a qualidade da cena exibida, o ser humano dificilmente seria capaz de absorver toda a informação exibida.

O mapeamento de dados categóricos com o atributo visual cor também é outra questão que limita o uso da GraphMiner. Um número máximo de 20 cores distintas é usado para visualmente apresentar os valores de atributos categóricos. Além disso, algumas funcionalidades básicas ainda não foram incluídas na ferramenta. Notadamente, faltam implementar as funcionalidades para imprimir, para rotular de conjuntos de registros na tela visual e para salvar consultas e mapeamentos em disco. Estas funcionalidades, bem como a possibilidade de escolher as cores utilizadas no mapeamento de dados categóricos serão implementadas futuramente na GraphMiner.

O número de controles de consulta existentes na ferramenta é insuficiente. A barra de intervalo é o único controle de consulta implementado. Existem outros controles como caixa de seleção e botões de opção que podem ser implementados na GraphMiner. As caixas de seleção, exibidas na Figura 6.13 (a), permitem que os atributos sejam selecionados um a um em qualquer combinação. Se a variável mostrada no controle estiver sendo usada como variável de coloração, as cores em que cada atributo é representado são mostradas ao lado do mesmo. Os botões de opção mostrados na Figura 6.13 (b) permitem que os atributos sejam selecionados de forma exclusiva. Da mesma forma que nas caixas de seleção as cores são mapeadas no controle [Almeida, 2003].



**Figura 6.13.** Controles visuais com caixas de verificação (a) e botões de opção (b) [Almeida, 2003].

## 6.4 CONSIDERAÇÕES FINAIS

A utilização de bancos de dados relacionais no processo de armazenamento de informações é algo cada vez mais comum nas organizações modernas. A aplicação de mecanismos de visualização neste contexto permite uma melhor representação das estruturas informais dos dados, possibilitando ainda a identificação de possíveis pontos fortes a serem explorados.

A aplicação da ferramenta GraphMiner nos contextos citados foi importante para a percepção de que, mesmo em situações diferentes, os processos de descoberta de informações em ambos exemplos de uso puderam ser apoiados pelo seu uso. Os exemplos mostraram que a ferramenta possui flexibilidade para se adequar a diferentes contextos e, portanto, é independente de tecnologia ou domínio específico. Ao mesmo tempo, foi constatado que novas funcionalidades podem ser incorporadas pela ferramenta, como a permitir que o usuário salve o estado do grafo e possa recompor a cena visual sem necessariamente ter que repetir todo o processo de mineração visual de dados e criar mecanismos que permitam selecionar ocultar ou mostrar múltiplos nós em uma cena.

Neste capítulo foram apresentados dois exemplos de uso com a ferramenta Graphminer, como uma forma de subsidiar a avaliação de sua aplicabilidade em contextos diversos. No próximo capítulo será apresentada a conclusão desta dissertação, incluindo considerações sobre trabalhos futuros e a evolução da ferramenta GraphMiner.

# CONCLUSÕES

Atualmente, os conjuntos de dados relacionais são uma das principais fontes de armazenamento de dados estruturados. Isto inclui dados científicos oriundos de experimentos diversos, dados comerciais em sistemas de gerenciamento de bancos de dados e dados contidos em documentos hipermídia. A exploração analítica destes dados normalmente requer a participação humana para obter entendimento maior sobre a massa de dados. Neste caso, um dos problemas genéricos a serem resolvidos é o de explorar visualmente esses dados. A solução para este problema tem aplicações imediatas em áreas como a mineração de dados, a classificação e recuperação de informações e ainda, a navegação sobre bases de dados relacionais.

Considerando a importância da visualização de informações para as mais diversas áreas do conhecimento humano, este trabalho apresentou uma abordagem para a visualização exploratória de dados contidos em bancos de dados relacionais. Esta abordagem engloba a definição de uma ferramenta para a visualização de relacionamentos entre informações de diferentes domínios e do mapeamento de uma forma de dados relacionais para esta ferramenta. A ferramenta se destaca pelo fato de permitir o mapeamento e a visualização de dados de maneira intuitiva, apoiando o processo de descoberta de conhecimento a partir de dados relacionais.

## 7.1 CONTRIBUIÇÕES

O trabalho disponibilizou uma ferramenta que apóia o processo de descoberta de conhecimento através da representação visual dados contidos em conjuntos de dados relacionais e seus relacionamentos.

Dentre as principais contribuições deste trabalho pode-se destacar:

- i) A dissertação dá uma visão geral do cenário atual de modelos e ferramentas para visualização de informações baseada na metáfora visual de um grafo.
- ii) Uma abordagem de mapeamento de dados armazenados em bancos de dados relacionais para grafos.
- iii) A interface visual e navegacional da ferramenta é reutilizável em aplicações específicas de domínios específicos.
- iv) A descrição e avaliação da experiência de aplicação da ferramenta para visualização de informações nos dois exemplos de uso.

Os benefícios do trabalho desenvolvido puderam ser avaliados durante a pesquisa bibliográfica e revisão da literatura, os processos de modelagem e implementação da ferramenta GraphMiner, bem como ao longo do desenvolvimento dos exemplos de uso.

## 7.2 TRABALHOS FUTUROS

As limitações discutidas na seção 6.3 precisam ser solucionadas. Entre elas podemos citar: (1) propor a persistência e navegação sobre grandes volumes de dados; (2) desenhar e atualizar dinamicamente grafos complexos.

Além das limitações, os exemplos de uso apresentados no decorrer deste trabalho apontaram novas funcionalidades que podem vir a ser acrescentadas à ferramenta em trabalhos futuros. A principal extensão sugerida foi que o aplicativo permita salvar os objetos que compõem o grafo na cena visual. Esta funcionalidade é especialmente útil para situações que envolvam consultas complexas sobre os dados, pois permitiria ao usuário salvar consultas e mapeamentos visuais complexos e reusá-las quando necessário.

Uma outra necessidade apontada durante os exemplos de uso foi a disponibilização de um controle que permita a seleção e o agrupamento ou expansão de múltiplos nós do grafo. O usuário poderia focar a visualização em uma determinada área do grafo amenizando o problema da visualização de grafos com muitos nós. Além disso, pode-se relacionar alguns outros trabalhos futuros relacionados à evolução da ferramenta Graph-Miner:

- i) Disponibilização de novas formas de acesso a diferentes fontes de dados, como servidor de documentos e sistemas de arquivos. Atualmente a ferramenta disponibiliza diversas funcionalidades para acesso a bancos de dados relacionais.
- ii) Adoção de uma tabela de cores dependendo do domínio da aplicação.
- iii) A visualização de informações em contextos específicos como ontologias a partir de documentos RDF [W3C, 2004], documentos em portais corporativos e informações retornadas de mecanismos de buscas na Web.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [A. Silberschatz and Sudarshan, 1999] A. Silberschatz, H. F. K. and Sudarshan, S. (1999). *Sistemas de bancos de dados*. Makron.
- [Adar and Tyler, 2003] Adar, E. and Tyler, J. (2003). Zoomgraph. Technical report, Information Dynamics Lab, HP Laboratories CA USA.
- [aiSee, 2004] aiSee (2004). aisee graph visualization. <http://www.aisee.com>.
- [Almeida, 2003] Almeida, M. (2003). Uma ferramenta para mineração visual de dados usando mapas em árvore e suas aplicações. Master's thesis, Unifacs.
- [Amazon, 2004] Amazon (2004). Touchgraph amazonbrowser. <http://www.touchgraph.com/TGAmazonBrowser.html>.
- [Ankerst, 2000] Ankerst, M. (2000). *Visual Data Mining*. PhD thesis.
- [B. A. Price and Small, 1998] B. A. Price, R. B. and Small, I. (1998). Software visualization. Technical report, Massachusetts Institute of Technology and Cambridge, London.
- [Batagelj and Mrvar, 1997] Batagelj, V. and Mrvar, A. (1997). Pajek, program for large analysis. <http://www.vlado.fmf.uni-lj.si/pub/networks/pajek>.
- [Branco, 2003] Branco, V. M. A. (2003). Visualização como suporte a exploração de uma base de dados pluviométricos. Master's thesis, Programa de Pós-Graduação em Ciências da Computação e Matemática Computacional-USP São Carlos.
- [C. Duncan and Kobourov, 1998] C. Duncan, M. G. and Kobourov, S. (1998). Balanced aspect ratio trees and their use for drawing very large graphs. *Lecture Notes in Computer Science*.
- [C. M. D. S. Freitas M. O. Chubachi and Cava, 2001] C. M. D. S. Freitas M. O. Chubachi, P. R. G. L. and Cava, R. A. (2001). Introdução à visualização de informações. *Revista de Informática Teórica e Aplicada*, 8(2):143–158.
- [Cherno, 1973] Cherno, H. (1973). The use of faces to represent points in kdimensional space graphically. *Statistical Analises*, 68:361–368.
- [Chiavenatto, 1992] Chiavenatto, I. (1992). *Recursos Humanos*. Atlas.
- [D. A. Keim and Seidl, 1994a] D. A. Keim, H. P. K. and Seidl, T. (1994a). Supporting data mining of large databases by visual feedback queries. In *Proc. 10th Int'l Conf. Engineering, Houston, Texas*, pages 302–313.
- [D. A. Keim and Seidl, 1994b] D. A. Keim, H. P. K. and Seidl, T. (1994b). Visdb: Database exploration using multidimensional visualization. pages 40–49.



- [D. Krackhardt and Mcgrath, 1995] D. Krackhardt, J. B. and Mcgrath, C. (1995). Krackplot 3.0 user's manual. Technical report, Analytic Technologies.
- [Date, 1986] Date, C. J. (1986). *Introdução à sistemas de bancos de dados*. Campos.
- [Eades, 1984] Eades, P. (1984). A heuristic for graph-drawing. In *Congressus Numerantium*, pages 42:1 49–160.
- [Eades, 1992] Eades, P. (1992). Drawing free trees. In *Bulletin of the Institute for Combinatorics and its Application*.
- [Erlingsson and Krishnamoorthy, 1996] Erlingsson, U. and Krishnamoorthy, M. (1996). Interactive graph drawing on the world wide web. Technical report, Rensselaer Polytechnic Institute, NY USA.
- [Freitas and Wagner, 1995] Freitas, C. M. D. S. and Wagner, F. R. (1995). Ferramentas de suporte às tarefas da análise exploratória visual. *Revista de Informática Teórica e Aplicada*, 2(1):5–36.
- [G. D. Battista and Tollis, 1994] G. D. Battista, P. E. and Tollis, I. G. (1994). *Algorithms for drawing graphs: An annotated bibliography*. Computational Geometry, 4th edition.
- [G. D. Battista and Tollis, 1999] G. D. Battista, P. Eades, R. T. and Tollis, I. G. (1999). *Graph Drawing - Algorithms for the Visualization of Graphs*. Prentice Hall, 1th edition.
- [Gansner and North, 1999] Gansner, R. and North, S. (1999). An open graph visualization system and its applications to software engineering. Technical report, ATeT Labs - Research, Shannon Laboratory, NJ-USA.
- [Gdt, 2004] Gdt (2004). Graph drawing toolkit. <http://www.dia.uniroma3.it/gdt/index.html>.
- [Google, 2004] Google (2004). Touchgraph googlebrowser. <http://www.touchgraph.com/TGGoogleBrowser.html>.
- [Graphlet, 2004] Graphlet (2004). Toolkit for graph editors and graph algorithms. <http://infosun.fmi.uni-passau.de/Graphlet>.
- [Hadany and Harel, 1999] Hadany, R. and Harel, D. (1999). A multi-scale algorithm for drawing graphs nicely. *Graph Theoretic Concepts in Computer Science (WG '99)*.
- [Hoffman and Grinstein, 1999] Hoffman, P. and Grinstein, G. (1999). A survey of visualizations for high-dimensional data mining. pages 47–82.
- [Ilog, 2004] Ilog (2004). Ilog jviews graph layout package. <http://www.ilog.com/products/jviews/graphlayout/>.
- [Inselberg, 1997] Inselberg, A. (1997). Multidimensional detective. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, pages 100–107. IEEE.
- [Inselberg and Dimsdale, 1990] Inselberg, A. and Dimsdale, B. (1990). Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the 1990 IEEE Symposium on Information Visualization (InfoVis '90)*, pages 361–375. IEEE.

- [J. Branke and Schmeck, 1997] J. Branke, F. B. and Schmeck, H. (1997). Using genetic algorithms for drawing undirected graphs. In *Third Nordic Workshop on Genetic Algorithms and thier Applications*.
- [Kamada and Kawai, 1989] Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. In *Information Processing Letters*, pages 31:7–15.
- [Kartoo, 2004] Kartoo (2004). Kartoo technologies. <http://www.kartoo.net>.
- [Keim, 2000] Keim, D. A. (2000). Designing pixel-oriented visualization techniques: Theory and applications. 6(5):59–78.
- [Keim, 2001] Keim, D. A. (2001). Visual exploration of large data sets. 44(8):38–44.
- [Keim, 2002] Keim, D. A. (2002). Information visualization and visual data mining. 8(1):1–8.
- [Keim and Kriegel, 1995] Keim, D. A. and Kriegel, H. P. (1995). Issues in visualizing large databases. In *Proc. Conf. on Visual Database Systems (VDB), Lausanne, Schweiz, 1995, In: Visual Database Systems*.
- [Keim and Kriegel, 1996] Keim, D. A. and Kriegel, H. P. (1996). Visualization techniques for mining large databases: A comparison. 8(6):923–938.
- [Knuth, 1963] Knuth, D. E. (1963). Computer drawn flowchart.
- [Kruskal and Seery, 1980] Kruskal, J. B. and Seery, J. B. (1980). Designing network diagrams. In *In Proc. First general Conference on Social Graphics. U.S. Department of the Census*.
- [Leeuw and Michailides, 1999] Leeuw, J. and Michailides, G. (1999). Graph layout techniques and multidimensional data analysis. Technical report, UCLA.
- [Libsea, 2003] Libsea (2003). The libsea graph file format and java graph library. <http://www.caida.org/tools/visualization/libsea/>.
- [Livejournal, 2004] Livejournal (2004). Touchgraph livejournal browser. <http://www.touchgraph.com>.
- [M. S. Marshall and Malancon, 2000] M. S. Marshall, I. H. and Malancon, G. (2000). An object-oriented design for graph visualization. Technical report, CWI Technical Report.
- [Marques and Mendonça, 2003] Marques, M. C. and Mendonça, M. G. M. (2003). Graph-miner: Ferramenta de mineração visual baseada em grafos.
- [Mendonça, 2001] Mendonça, M. (2001). Mineração de dados. 1:189–218.
- [N.Conklin, 2004] N.Conklin (2004). Information visualization publication ranking. <http://infovis.cs.vt.edu/snap/projects/nathan.snap/papers/infovis.pub.ranking.pdf>.
- [Niggemann, 2001] Niggemann, O. (2001). *Visual Data Mining of Graph-Based Data*. PhD thesis.

- [P. Eades and Huang, 1997] P. Eades, R. C. and Huang, M. (1997). Online animated graph drawing for web navigation. In *In Proc. Graph Drawing 97. Springer*.
- [P. Hoffman and Stanley, 1997] P. Hoffman, G. Grinstein, K. M. I. G. and Stanley, E. (1997). Dna visual and analytic data mining. *IEEE Journal*.
- [Pickett and Grinstein, 1988] Pickett, R. M. and Grinstein, G. G. (1988). Iconographic displays for visualizing multidimensional data. In *In Proc. IEEE Conf. on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ*, pages 514–519.
- [Planet, 2004] Planet (2004). Planet-wissen navigator. <http://www.touchgraph.com>.
- [Roxborough and Arunabha, 1996] Roxborough, T. and Arunabha, A. (1996). Graph clustering using multiway ratio cut. in s. north, editor, graph drawing, lecture notes in computer science. Technical report, Springer Verlag.
- [S. Guha and Shim, 1998] S. Guha, R. R. and Shim, K. (1998). Cure: An efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*.
- [S. K. Card and Shneiderman, 1999] S. K. Card, J. D. M. and Shneiderman, B. (1999). *Information Visualization. In: Readings in Information Visualization - Using Visualization to Think*. Morgan Kaufmann.
- [Sablowski and Frick, 1996] Sablowski, R. and Frick, A. (1996). Automatic graph clustering. in s. north, editor, graph drawing, lecture notes in computer. Technical report, Springer Verlag.
- [S.Borgatti, 2004a] S.Borgatti (2004a). Netdraw, analytic technologies. <http://www.analytictech.com/downloadnd.htm>.
- [S.Borgatti, 2004b] S.Borgatti (2004b). Ucinet, analytic technologies. <http://www.analytictech.com/downloadnd.htm>.
- [Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages, Boulder, CO*, pages 336–343.
- [SQL-92, 2004] SQL-92 (2004). Structured query language. <http://www.service-architecture.com/database/articles/sql-92.html>.
- [Sugiyama and Toda, 1981] Sugiyama, S. T. K. and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. 2(11).
- [Sun Microsystems, 2003] Sun Microsystems, I. (2003). The source for java technology. <http://java.sun.com>.
- [Szwarcfiter, 1986] Szwarcfiter, J. (1986). *Grafos e Algoritmos Computacionais*. Campus.
- [ThinkMap, 2004] ThinkMap (2004). Thinkmap visualization interface. <http://www.plumbdesign.com/products/thinkmap>.
- [Togerson, 1952] Togerson, W. (1952). Multidimensional scaling: I. theory and method. Technical report, Psychometrika.

- [TouchGraph, 2004] TouchGraph (2004). Touchgraph llc. <http://www.touchgraph.com>.
- [Tufte, 1983] Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press.
- [Tufte, 1990] Tufte, E. R. (1990). *[0]Envisioning Information*. Graphics Press.
- [W. Chung and Jr, 2002] W. Chung, H. C. and Jr, J. N. (2002). Business intelligence explorer: A knowledge map framework for discovering business intelligence on the web. In *Proceedings of 36th Hawaii International Conference on System Sciences*.
- [W3C, 2004] W3C (2004). Rdf/xml syntax specification. <http://www.w3.org/TR/rdf-syntax-grammar>.
- [Ward, 1994] Ward, M. O. (1994). Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *in Proc. Visualization 94, Washington, DC*, pages 326–336.
- [Wegman, 1990] Wegman, E. (1990). Hyperdimensional data analysis using parallel coordinates. *American Statistical Association*, 85(11):664–675.
- [Wegman and Luo, 1996] Wegman, E. J. and Luo, Q. (1996). High dimensional clustering using parallel coordinates and the grand tour. Technical Report 124, Center for Computational Statistics, George Mason University.
- [WikiBrowser, 2004] WikiBrowser (2004). Tg wikibrowser. <http://www.touchgraph.com>.
- [Wong and Bergeron, 1997] Wong, P. and Bergeron, R. (1997). 30 years of multidimensional multivariate visualization, scientific visualization-overviews, methodologies and techniques. pages 3–33.
- [yWorks, 2004] yWorks (2004). yworks. <http://www.yworks.de/>.